UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 1. stopnja

Katja Kuster

Optimalna pot na goro

Delo diplomskega seminarja

Mentor: izred. prof. dr. Gašper Jaklič
Somentor: dr. Tadej Kanduč

Ljubljana, 2015

Kazalo

Optimalna pot na goro

Povzetek

Diplomski seminar obravnava problem iskanja krivulje na ploskvi, ki zadošča dodatnim zahtevam. Posvetili se bomo zanimivemu primeru iskanja optimalne poti na goro in se osredotočili na enega izmed možnih kriterijev optimalnosti. V našem primeru bo to minimalna poraba energije, ki jo za vzpon potrebuje povprečen pohodnik.

Najprej bomo iz višinskih točk pobočja gore z uporabo gladkih polinomskih zlepkov konstruirali relief. Nato bomo izračunali porabo energije na vsaki izmed robnih krivulj manjših koščkov reliefa (krp). S seštevkom energij bomo dobili porabo energije vzdolž poljubne poti, od izhodišča pa do vrha gore. Za iskanje optimalne poti bomo uporabili Dijkstrov algoritem in tako optimizacijski problem prevedli na diskretni problem iskanja najcenejše poti na mreži polinomskih krivulj.

Na koncu si bomo podrobneje ogledali predstavljene algoritme in postopek iskanja optimalne poti na goro preverili na konkretnih primerih.

Optimal Mountain Ascent

Abstract

This work focuses on a problem of finding a curve on the surface with some additional requirements. We will look into an interesting example of finding an optimal mountain ascent and focus on one of the possible criteria of optimality. In our case, this will be the minimal amount of energy consumption for an average hiker.

First we will construct a terrain description using smooth polynomial splines from the given data. Then we will calculate the energy consumption along each of the boundary curves of the spline patches. With the sum of all energy values we will get the energy consumption along the path from the starting point to the summit of the mountain. We will rewrite the original problem of finding an energy minimizing mountain ascent as a search for the cheapest path in a network and solve it by Dijkstra's algorithm.

At the end we will take a closer look at how to use the presented algorithm for finding an optimal mountain ascent in practice.

Math. Subj. Class. (2010): 41A05, 41A15, 65D17

Ključne besede: vzpon na goro, aproksimacija, poraba energije

Keywords: mountain ascent, approximation, energy consumption

1. Uvod

Iskanje optimalne poti na goro je poseben primer iskanja krivulje med dvema danima točkama na ploskvi pod nekimi dodatnimi pogoji. Pri določanju algoritma, ki bi kot rezultat vrnil želeno krivuljo, se soočimo z več težavami. Zaradi razgibanega naravnega terena moramo najprej poiskati dober način predstavitve ploskve in zapis iskane krivulje, vložene na ploskev. Naslednji korak pa je izbira primernega kriterija optimalnosti in izbira podprostora za iskanje rešitve (diskretizacija).

Omenjeni problem je v praksi precej uporaben, pogosto ga srečamo v gradbeništvu. Pri načrtovanju nove ceste ali železnice je namreč potrebno paziti na omejitve naklona in radija ovinkov ter upoštevati geomorfološke lastnosti terena. Da bo cesta zgrajena v funkcionalnih okvirih, ne sme biti prestrma, pomemben dejavnik pa je tudi cena gradnje, ki se z vključevanjem mostov ali tunelov znatno poveča.

Po svetu najdemo ogromno zanimivih cestnih odsekov, ki velike naklone terena premagujejo v obliki cikcak vzorcev. Primer take ceste vidimo na sliki 1.

Slika 1. Cesta skozi sotesko Todra Gorge v Maroku.

V diplomskem seminarju bomo proučili nekoliko enostavnejši primer s področja pohodništva in sicer načrtovanje nove planinske poti. Dobra planinska pot je tista, ki se izogne težko prehodnemu terenu in uporablja naravne prehode. Ugodno je, da ne izgublja višine, je čim krajša in, zaradi nevarnosti zdrsa, ne preveč strma. V nalogi se bomo osredotočili na enega izmed možnih kriterijev za iskanje optimalne poti – minimalno porabo energije.

Najprej bomo predstavili interpolacijo danih podatkov s primerno ploskvijo, nato pa s pomočjo energijskega funkcionala izračunali porabo energije, ki jo potrebujemo za vzpon. Z uporabo Dijkstrovega algoritma bomo iz velike množice možnih poti na goro poiskali optimalno. Za boljše razumevanje predstavljenih postopkov si bomo na koncu ogledali še konkretna primera iskanja optimalne poti na Šmarno goro ter na Triglav.

2. Interpolacija

Prvi korak pri iskanju optimalne poti na goro je predstavitev ploskve in zapis krivulje, vložene na ploskev. Iz danih podatkov želimo aproksimirati teren, tako da bo čim bolj podoben naravnemu reliefu.

V praksi dobimo mrežo geodetskih (GPS) podatkov, običajno so to točke v prostoru $(x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, 2, \ldots, n$. Najprej želimo poiskati interpolacijsko ploskev, ki bo potekala skozi dane točke.

Oglejmo si problem nekoliko podrobneje. Z Ω označimo konveksno ovojnico danih točk $v_i := (x_i, y_i)$ v ravnini.

Definicija 2.1. *Interpolacijska funkcija* je funkcija

$$f : \Omega \subset \mathbb{R}^2 \to \mathbb{R},$$

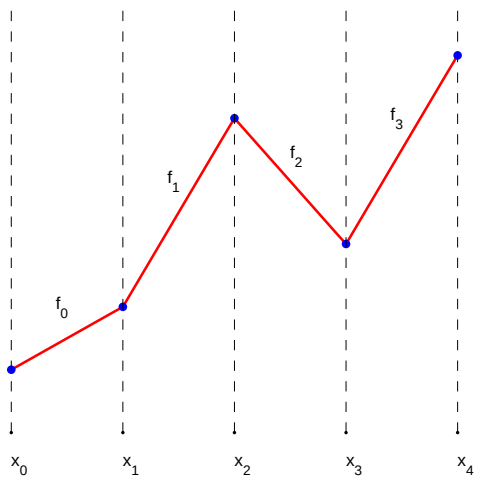
za katero velja, da je $f(v_i) = z_i$ za vsak $i = 1, 2, \ldots, n$.

2.1. **Linearna interpolacija.** Običajna rešitev problema bi bil zapis v obliki odsekoma linearne ploskve, saj sta njena konstrukcija in delo z njo enostavna. V ta namen si oglejmo preprost ravninski primer.

Primer 2.2. Denimo, da imamo dane točke $x_i \in \mathbb{R}$, $i = 0, 1, \ldots, n$. Iščemo interpolacijsko funkcijo, ki bo v teh točkah po vrsti zavzela vrednosti y_i, $i = 0, 1, \ldots, n$. Najpreprostejši način za zapis iskane funkcije je *odsekoma linearna interpolacija*. Interval $[x_0, x_n]$ razdelimo na podintervale $[x_i, x_{i+1}]$ za $i = 0, 1, \ldots, n-1$. Na vsakem izmed podintervalov dane podatke interpoliramo z linearno funkcijo

$$f_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i).$$

Na sliki 2 je prikazan primer odsekoma linearne interpolacije na petih točkah.



Slika 2. Zgled linearne interpolacije za $n = 4$.

Ker interpolacijska funkcija poteka skozi točke (x_i, y_i), $i = 0, 1, \ldots, n$ in v notranjih točkah (x_i, y_i), $i = 1, 2, \ldots, n-1$, velja

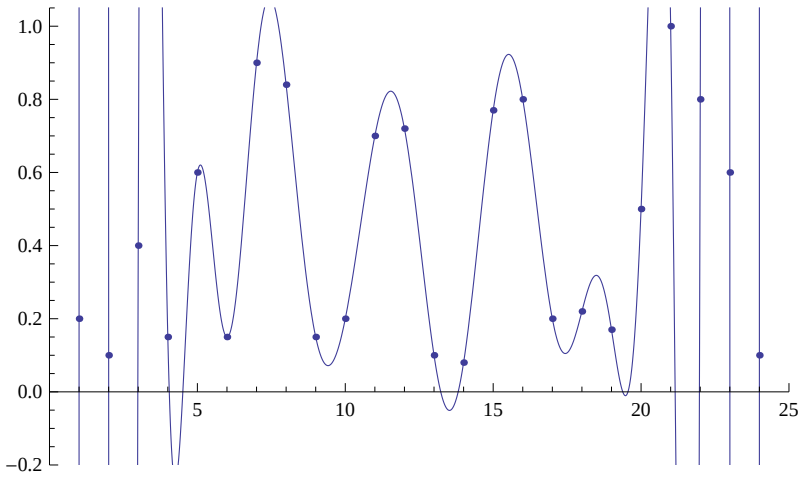
$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}) = y_{i+1},$$

je dobljena odsekoma linearna interpolacijska funkcija zvezna. Seveda pa ni nujno zvezno odvedljiva.

Na podoben problem naletimo pri odsekoma linearni interpolaciji s ploskvijo v prostoru. Preveliko število podatkov, ki so potrebni za dobro aproksimacijo terena, in dejstvo, da na ta način dobimo le zvezno ploskev, nas vodita do iskanja drugih možnosti za zapis interpolacijske ploskve.

2.2. **Polinomska interpolacija.** Predmet našega proučevanja so krivulje, vložene na ploskev, ki predstavljajo pohodne poti. Da se bo oblika dobljenih poti čim bolj približala naravnim, je *polinomska interpolacija* najbolj naravna izbira metode.

Ponovno si najprej oglejmo preprostejši primer v ravnini. Za $i = 0, 1, \ldots, n$ imamo dane paroma različne točke $x_i \in \mathbb{R}$ in vrednosti funkcije f v danih točkah, $y_i \in \mathbb{R}$. Poiskati želimo *interpolacijski polinom*, to je polinom p stopnje manjše ali enake n, ki se s funkcijo f ujema v točkah $x_0, \ldots, x_n$. Tu naletimo na problem, ko polinom slabo opiše interpolacijske podatke. Pri polinomih visokih stopenj se namreč lahko zgodi, da interpolacijski polinom preveč oscilira in se zato slabo prilega originalni krivulji. Primer takega polinoma vidimo na sliki 3. Omenjeni slabosti se lahko izognemo z uporabo *odsekoma polinomskih funkcij - zlepkov.*

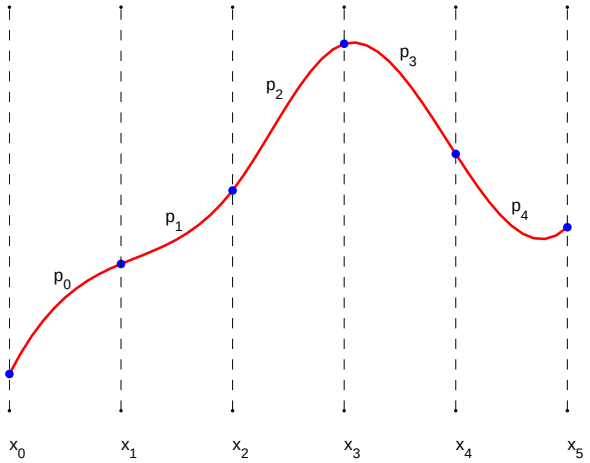


Slika 3. Primer, ko interpolacijski polinom preveč oscilira.

Definicija 2.3. *Zlepki* (angl. *splines*) so razred odsekoma polinomskih funkcij.

Spomnimo se primera 2.2, ko smo na podoben način definirali odsekoma linearno funkcijo. Ponovno imamo dane točke $x_i \in \mathbb{R}$, $i = 0, 1, \ldots, n$. Poiskati želimo interpolacijsko funkcijo (v tem primeru zlepek), ki bo po vrsti zavzela vrednosti y_i, $i = 0, 1, \ldots, n$. Kot je razvidno s slike 4, interval $[x_0, x_n]$ razdelimo na podintervale $[x_i, x_{i+1}]$ za $i = 0, 1, \ldots, n-1$. Na vsakem izmed podintervalov poiščemo interpolacijski polinom p_i nizke stopnje, tako da velja $p_i(x_i) = y_i$ in $p_i(x_{i+1}) = y_{i+1}$. Kot v linearnem primeru je seveda tudi tu dobljena funkcija zvezna, saj v notranjih točkah velja $p_i(x_{i+1}) = p_{i+1}(x_{i+1}) = y_{i+1}$. Če iz množice vseh možnih zlepkov, ki interpolirajo točke (x_i, y_i), izberemo pravega, lahko dosežemo tudi zvezno odvedljivost. Osredotočili se bomo na kubične zlepke, ki jih v praksi srečamo najpogosteje.

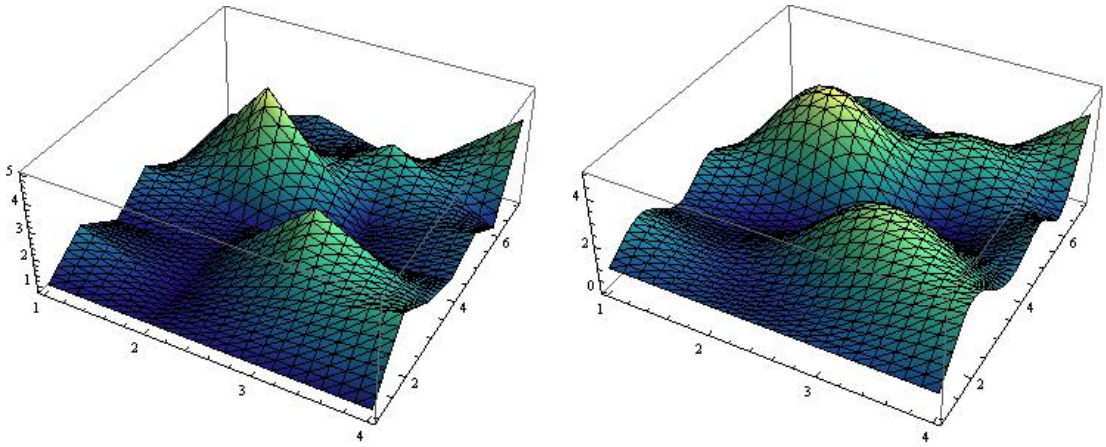
Primer 2.4. Na sliki 4 je prikazan primer odsekoma polinomske interpolacije na šestih točkah s kubičnim zlepkom f. Na vsakem izmed podintervalov $[x_i, x_{i+1}]$ je polinom p_i stopnje manjše ali enake tri.



Slika 4. Zgled odsekoma polinomske interpolacije za $n = 5$.

Pravkar opisani ravninski zgled nam služi kot model, ko preidemo na višje dimenzije. Pri interpolaciji v prostoru se bomo zgledovali po opisanem primeru in namesto linearnih uporabili odsekoma polinomske ploskve (zlepke) višjih stopenj.

Zaradi večjega števila prostih parametrov bomo dosegli, da bo dobljena ploskev vsaj zvezno odvedljiva, krivulje na njej, ki predstavljajo planinske poti, pa bodo posledično bolj naravnih oblik. Na sliki 5 lahko primerjamo izgled reliefa glede na obliko interpolacijske ploskve.



Slika 5. Odsekoma linearna ploskev (levo) in kubični zlepek (desno).

3. Dvorazsežni zlepki

Interpolacijo s polinomi, predstavljeno v prejšnjem razdelku, bi sedaj radi posplošili na ploskve. Kot smo že zapisali, želimo poiskati zvezno odvedljivo ploskev, ki bo potekala skozi dane točke v prostoru. Najprej si oglejmo nekaj definicij, ki jih

potrebujemo za razumevanje naslednjega poglavja.

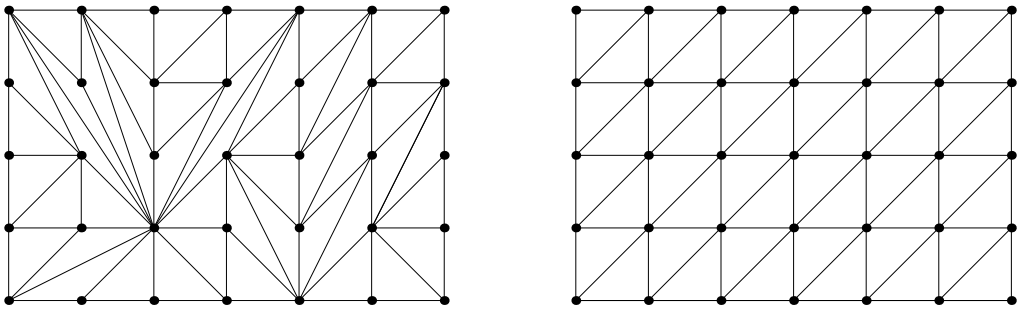
Naj bo z Ω, kot prej, označena konveksna ovojnica točk $v_i = (x_i, y_i) \in \mathbb{R}^2$.

Definicija 3.1. *Triangulacija* območja Ω je razdelitev območja na trikotnike z oglišči v točkah v_i.

Definicija 3.2. Za triangulacijo rečemo, da je *regularna*, če je vsak neprazen presek dveh različnih trikotnikov iz triangulacije bodisi skupno oglišče bodisi skupna stranica.

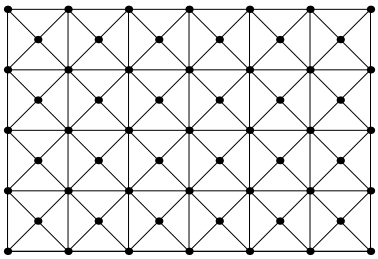
Slika 6. Primera neregularne (levo) in regularne triangulacije (desno).

Triangulacija na točkah v_i ni enolična in ima, kot bomo videli kasneje, velik vpliv na obliko interpolacijske ploskve.



Slika 7. Primera dveh različnih triangulacij istega območja.

V nalogi bomo uporabili regularno triangulacijo tipa 2, prikazano na sliki 8. Zaradi simetrične oblike triangulacije ne pride do favoriziranja gibanja po levi ali desni diagonalni smeri. Zaradi še pomembnejšega zagotavljanja numerične stabilnosti pa je ugodno, da ima triangulacija povezane bližnje točke, notranje kote pa ne premajhne.



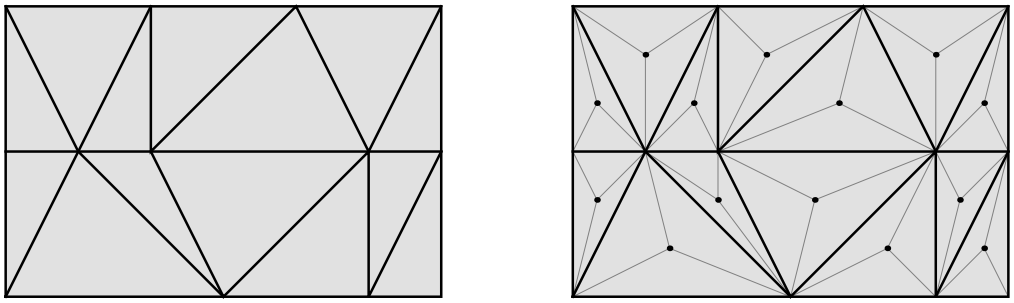
Slika 8. Primer triangulacije tipa 2.

Pri iskanju optimalne poti na goro vseskozi težimo k čim natančnejšim rezultatom. S tem namenom uvedemo pojem finejše triangulacije. Določanja finejše triangulacije se poslužimo takrat, ko želimo povečati število točk triangulacije.

Definicija 3.3. Naj bosta Δ in Δ_R triangulaciji danega območja Ω. Če velja, da je vsaka točka iz Δ tudi v Δ_R in da je vsak trikotnik iz Δ_R podtrikotnik nekega trikotnika iz Δ, potem rečemo, da je Δ_R *finejša triangulacija* od Δ.

Poznamo več učinkovitih načinov *subdivizije*, tj. razbitja trikotnikov triangulacije (*makro trikotnikov*) na več manjših trikotnikov (*mikro trikotnikov*). Spodaj je zapisan eden od možnih postopkov, ki ga bomo uporabili v poglavju 7.

Definicija 3.4. Dan je trikotnik $T := \langle v_1, v_2, v_3 \rangle$ s težiščem $v_T := (v_1 + v_2 + v_3)/3$. Točko v_T povežemo z vsemi oglišči trikotnika T in tako T razbijemo na tri manjše trikotnike. Finejši triangulaciji, ki jo dobimo na ta način, pravimo *Clough-Tocherjeva finejša triangulacija.*



Slika 9. Primer triangulacije (levo) in njene Clough-Tocherjeve finejše triangulacije (desno).

Z uvedbo triangulacije smo domeno Ω razdelili na trikotnike podobno, kot smo v prejšnjem poglavju interval razdelili na manjše podintervale. Sedaj nas čaka naslednji korak. Nad dano triangulacijo $\Delta := \{T_1, T_2, \ldots, T_N\}$ območja Ω želimo konstruirati gladko interpolacijsko funkcijo, za katero bo veljalo, da je njena zožitev na vsakega izmed trikotnikov triangulacije Δ dvorazsežni polinom predpisane stopnje. Razred takih funkcij bomo imenovali *zlepki*. Robove triangulacije bomo torej uporabili za iskanje optimalnih poti.

Pri konstrukciji omenjenih zlepkov naletimo na težave že povsem na začetku, saj sta obstoj in enoličnost interpolanta postavljena pod vprašaj. Pogoji, ki določajo gladkost funkcij, se praviloma med seboj tesno prepletajo in jih velikokrat ni mogoče izpolniti. S tem je tesno povezan tudi problem dimenzije prostora zlepkov, ki ostaja odprt. Poleg tega gre za reševanje velikega sistema linearnih enačb, na obliko zlepka pa ima lahko velik vpliv že sprememba enega samega interpolacijskega podatka. Ker želimo zagotoviti odvisnost le od lokalnih podatkov, hkrati pa bi se radi izognili še ostalim naštetim težavam, za reševanje problema uporabimo posebno vrsto zlepkov, *makroelemente*.

Makroelementi so posebna vrsta gladkih polinomskih zlepkov, zgrajenih nad dano triangulacijo ravninskega območja. Te odsekoma polinomske funkcije se zlepijo gladko, običajno vsaj enkrat zvezno odvedljivo, konstrukcija polinomskih delov pa je

lokalna. Polinomska funkcija, ki določa del makroelementa nad poljubnim trikotnikom triangulacije, je določena zgolj na podlagi podatkov nad tem trikotnikom. Za konstrukcijo sosednjih polinomov nad skupno stranico dveh trikotnikov uporabimo enake podatke. Prednost makroelementov, ki sicer velja za vse polinomske krpe, je tudi enostaven in učinkovit postopek subdivizije. Z delitvijo pripadajoče polinomske ploskve dosežemo povečevanje števila robnih krivulj makroelementa in s tem večjo natančnost dobljene optimalne poti.

V naslednjem poglavju si bomo natančneje ogledali bistvene lastnosti in trditve, ki veljajo za omenjene zlepke. V naslednjem poglavju si bomo natančneje ogledali bistvene lastnosti in trditve, ki veljajo za omenjene zlepke.

3.1. **Polinomi dveh spremenljivk.** V prejšnjem poglavju je bilo govora le o polinomih ene spremenljivke. Ker pa gre pri temi diplomske naloge za ploskve in krivulje v prostoru, si moramo ogledati še interpolacijske polinome dveh spremenljivk.

Definicija 3.5. *Polinom dveh spremenljivk stopnje d* definiramo kot

$$p(x,y) := \sum_{0\le i+j\le d} c_{ij}x^i y^j,$$

pri čemer je $d \in \mathbb{N}_0$ in $c_{ij} \in \mathbb{R}$.

Stopnja polinoma dveh spremenljivk je enaka maksimalni vsoti potenc obeh spremenljivk v posameznem monomu. S $\mathcal{P}_d$ označimo vektorski prostor takih polinomov stopnje manjše ali enake d. Njegovo bazo sestavljajo monomi oblike $\{x^i y^j\}_{0\le i+j\le d}$. Iz leksikografske ureditve monomov

$$(1) \qquad 1, \quad x, y, \quad x^2, xy, y^2, \quad \ldots, \quad x^d, x^{d-1}y, x^{d-2}y^2, \ldots, x^2y^{d-1}, xy^{d-1}, y^d$$

sledi, da je dimenzija prostora $\mathcal{P}_d$ enaka $1+2+\ \cdots\ +(d+1) = \frac{(d+1)(d+2)}{2} = \binom{d+2}{2}$.

Definicija 3.6. Množica

$$S_d^r(\Delta) := \{s \in C^r(\Omega) : s|_{T_i} \in \mathcal{P}_d, i = 1, \ldots, N\}$$

je *prostor odsekoma polinomskih funkcij oz. zlepkov* ($\mathcal{C}^r$ gladkih krivulj), pri čemer je $\Delta = \{T_1, \ldots, T_N\}$ triangulacija območja Ω, $\mathcal{P}_d$ pa prostor dvorazsežnih polinomov stopnje manjše ali enake d.

V prejšnjem poglavju smo definirali interpolacijski polinom v ravnini, sedaj pa se posvetimo konstrukciji interpolacijskega polinoma dveh spremenljivk.

Naj bodo dane paroma različne točke $v_i := (x_i, y_i) \in \mathbb{R}^2$ območja Ω. Polinom $p \in \mathcal{P}_d$ bi radi interpolirali skozi vrednosti $z_i \in \mathbb{R}$, tako da bo $p(v_i) = z_i$ za $i = 1, \ldots, n$. Pri tem je $n = \binom{d+2}{2}$. Bistvena razlika z enodimenzionalno interpolacijo je v tem, da tu polinom ne obstaja za poljubne točke v_i. Označimo s $\{h_j\}_{j=1}^n$ leksikografsko urejeno množico monomov (1). Dobimo zvezo

$$\sum_{j=1}^n c_j h_j(v_i) = z_i, \quad i = 1, \ldots, n.$$

Obstoj in enoličnost interpolacijskega polinoma sta zagotovljena takrat, ko bo matrika

$$[h_j(v_i)]_{i,j=1}^n$$

nesingularna.

3.2. **Baricentrične koordinate.** Ta vrsta koordinat bo imela pomembno vlogo pri razumevanju algoritma za iskanje optimalne poti, saj so v primerjavi z običajnimi kartezičnimi koordinatami zaradi lokalnosti te bolj primerne za delo s polinomi dveh spremenljivk nad trikotniki triangulacije. Z uporabo baricentričnih koordinat dosežemo geometrijsko bolj enoten in enostaven zapis.

Definicija 3.7. Naj bo $T := \langle v_1, v_2, v_3 \rangle$ neizrojen trikotnik v ravnini, določen z oglišči $v_i = (x_i, y_i)$ za $i = 1, 2, 3$. Privzamemo, da je T pozitivno orientiran. Vrednostim b_1, b_2 in b_3, za katere veljata zvezi

$$v = b_1 v_1 + b_2 v_2 + b_3 v_3 \quad \text{in} \quad b_1 + b_2 + b_3 = 1$$

rečemo *baricentrične koordinate* točke v glede na trikotnik T.

Lema 3.8. *Vsako točko $v = (x, y) \in \mathbb{R}^2$ lahko enolično zapišemo v baricentričnih koordinatah.*

Dokaz. Poljubno točko v želimo zapisati v baricentričnih koordinatah, tj. poiskati želimo take vrednosti b_1, b_2, b_3, da bosta veljali enačbi $v = b_1 v_1 + b_2 v_2 + b_3 v_3$ in $b_1 + b_2 + b_3 = 1$. Označimo $v_i = (x_i, y_i)$ in zapišimo linearni sistem

$$\underbrace{\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}}_{=:A} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Za pozitivno orientiran trikotnik T v ravnini z oglišči v točkah v_1, v_2, v_3 brez težav izračunamo njegovo ploščino

$$P_T = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \frac{1}{2} \det(A^T) = \frac{1}{2} \det(A).$$

Ker je ploščina pozitivno orientiranega neizrojenega trikotnika v ravnini vselej pozitivna, bo tudi $\det(A)$ pozitivna. Torej je matrika A neizrojena, enolično rešitev pa lahko dobimo s pomočjo Cramerjevega pravila

$$b_1 = \frac{1}{\det(A)} \begin{vmatrix} x & x_2 & x_3 \\ y & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix} = \frac{1}{2P_T} \begin{vmatrix} x & x_2 & x_3 \\ y & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}$$

Analogno dobimo še preostali dve koordinati:

$$b_2 = \frac{1}{2P_T} \begin{vmatrix} x_1 & x & x_3 \\ y_1 & y & y_3 \\ 1 & 1 & 1 \end{vmatrix}, \quad b_3 = \frac{1}{2P_T} \begin{vmatrix} x_1 & x_2 & x \\ y_1 & y_2 & y \\ 1 & 1 & 1 \end{vmatrix}$$

□

3.3. **Bézierove krivulje.** Ta tip krivulj sta neodvisno razvila inženirja P. de Casteljau in P. Bézier za potrebe oblikovanja v avtomobilski industriji, danes pa predstavlja osnovno implementacijo krivulj in ploskev v računalništvu, v povezavi z industrijskim oblikovanjem in modeliranjem ter zabavno industrijo (filmi, 3D igre itd.). V poštev bodo prišle tudi v tej nalogi, saj bomo v tej obliki predstavili robne krivulje makroelementa. Zato je dobro, da si najprej ogledamo nekaj definicij in lastnosti Bézierovih krivulj.

Definicija 3.9. Dane so točke $c_i \in \mathbb{R}^3$ za $i = 0, 1, \ldots, d$. *Bézierova krivulja* je krivulja oblike

$$b^d(t) = \sum_{j=0}^{d} c_j B_j^d(t),$$

pri čemer so B_j^d Bernsteinovi bazni polinomi stopnje d, c_j pa pripadajoče Bézierove kontrolne točke.

Definicija 3.10. Za $t \in [0, 1]$ definiramo i-ti *Bernsteinov bazni polinom* stopnje d kot

$$B_i^d(t) = \binom{d}{i} t^i (1-t)^{d-i}.$$

Izrek 3.11. *Za Bernsteinove bazne polinome velja rekurzivna zveza*

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t). \tag{2}$$

Pri tem velja

$$B_0^0(t) = 1$$

in

$$B_j^n(t) = 0 \quad \text{za} \quad j \notin \{0, 1, \ldots, n\}.$$

Dokaz.

$$\begin{aligned} B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n-1}{i} t^i (1-t)^{n-i} + \binom{n-1}{i-1} t^i (1-t)^{n-i} \\ &= (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \end{aligned}$$

□

Izrek 3.12. *Odvod Bézierove krivulje lahko zapišemo kot*

$$\frac{\mathrm{d}}{\mathrm{d}t} b^d(t) = d \sum_{j=0}^{d-1} \Delta c_j B_j^{d-1}(t),$$

kjer je $\Delta c_j = c_{j+1} - c_j \in \mathbb{R}^3$.

Dokaz. Najprej izračunajmo odvod Bernsteinovega baznega polinoma

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} B_i^d(t) &= \frac{\mathrm{d}}{\mathrm{d}t} \binom{d}{i} t^i (1-t)^{d-i} \\ &= \frac{id!}{i!(d-i)!} t^{i-1}(1-t)^{d-i} - \frac{(d-i)d!}{i!(d-i)!} t^i (1-t)^{d-i-1} \\ &= \frac{d(d-1)!}{(i-1)!(d-i)!} t^{i-1}(1-t)^{d-i} - \frac{d(d-1)!}{i!(d-i-1)!} t^i (1-t)^{d-i-1} \\ &= d\,[B_{i-1}^{d-1}(t) - B_i^{d-1}(t)]. \end{aligned}$$

Odvod Bézierove krivulje je torej enak

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} b^d(t) &= d \sum_{j=0}^{d} c_j [B_{j-1}^{d-1}(t) - B_j^{d-1}(t)] \\
&= d \sum_{j=1}^{d} c_j B_{j-1}^{d-1}(t) - d \sum_{j=0}^{d-1} c_j B_j^{d-1}(t) \\
&= d \sum_{j=0}^{d-1} c_{j+1} B_j^{d-1}(t) - d \sum_{j=0}^{d-1} c_j B_j^{d-1}(t) \\
&= d \sum_{j=0}^{d-1} (c_{j+1} - c_j) B_j^{d-1}(t).
\end{aligned}
$$

□

Opazimo, da je odvod Bézierove krivulje spet Bézierova krivulja, pridobljena s kontrolnim poligonom iz diferenc točk prvotnega polinoma.

Razdelek o Bernsteinovih baznih polinomih bomo sedaj posplošili na primer polinomov dveh spremenljivk.

Definicija 3.13. Naj bo T trikotnik v ravnini, $v = (x, y)$ pa točka z baricentričnimi koordinatami (b_1, b_2, b_3). *Bernsteinov bazni polinom stopnje* d glede na trikotnik T definiramo kot

$$B_{ijk}^d(v) = \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k, \quad i + j + k = d.$$

Iz dokaza leme 3.8 sledi, da lahko baricentrične koordinate, definirane v prejšnjem podpoglavju (definicija 3.7) zapišemo kot

$$b_1 = \frac{(x_2 y_3 - y_2 x_3) + x(y_2 - y_3) + y(x_3 - x_2)}{2P_T}.$$

Analogno zapišemo koordinati b_2 in b_3. Opazimo, da je b_i za $i = 1, 2, 3$ linearni polinom spremenljivk x in y. Od tod očitno sledi, da je Bernsteinov polinom $B_{ijk}^d(x, y)$ res polinom stopnje d.

Definicija 3.14. Množico $\{B_{ijk}^d\}_{i+j+k=d}$ moči $\binom{d+2}{2}$ imenujemo Bernstein-Bézierova baza oz. krajše Bernsteinova baza.

Izrek 3.15. *Množica Bernsteinovih baznih polinomov*

$$\mathcal{B}^d = \{B_{ijk}^d\}_{i+j+k=d}$$

je baza prostora polinomov $\mathcal{P}_d$.

Dokaz. Moč množice Bernsteinovih baznih polinomov se ujema z dimenzijo prostora $\mathcal{P}_d$, torej nam preostane le še dokaz, da vsi bazni polinomi prostora $\mathcal{P}_d$ ležijo v linearni ogrinjači $\mathcal{L}(\mathcal{B}^d)$. Pri dokazu si bomo pomagali z lastnostjo Bernsteinovih polinomov, tj. da tvorijo particijo enote:

$$\sum_{i+j+k=d} B_{ijk}^d(v) = \sum_{i+j+k=d} \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k = (b_1 + b_2 + b_3)^d = 1.$$

Velja torej $1 \in \mathcal{L}(\mathcal{B}^d)$. Vzamemo poljubno točko $v = b_1v_1+b_2v_2+b_3v_3$ in jo zapišemo po komponentah

$$\begin{aligned}
x &= (b_1x_1 + b_2x_2 + b_3x_3) \cdot 1 \\
&= (b_1x_1 + b_2x_2 + b_3x_3) \cdot \sum_{i+j+k=d-1} B^{d-1}_{ijk}(x,y) \\
&= \sum_{i+j+k=d-1} (b_1x_1B^{d-1}_{ijk}(x,y) + b_2x_2B^{d-1}_{ijk}(x,y) + b_3x_3B^{d-1}_{ijk}(x,y)) \\
&= \sum_{i+j+k=d-1} \frac{1}{d}(x_1(i+1)B^d_{i+1,j,k} + x_2(j+1)B^d_{i,j+1,k} + x_3(k+1)B^d_{i,j,k+1}).
\end{aligned}$$

Če vsoto razbijemo na tri dele, zamaknemo indekse, nato pa ponovno združimo vse tri vsote dobimo

$$\begin{aligned}
x &= \sum_{i+j+k=d} \frac{1}{d}(ix_1B^d_{ijk} + jx_2B^d_{ijk} + kx_3B^d_{ijk}) \\
&= \sum_{i+j+k=d} \frac{1}{d}(ix_1 + jx_2 + kx_3)B^d_{ijk}.
\end{aligned}$$

Na podoben način pokažemo, da je

$$y = \sum_{i+j+k=d} \frac{1}{d}(iy_1 + jy_2 + ky_3)B^d_{ijk}.$$

Na ta način smo dokazali, da x in y ležita v linearni ogrinjači $\mathcal{L}(\mathcal{B}^d)$.

Ker želimo dokazati, da to velja za poljuben bazni polinom prostora $\mathcal{P}_d$, uporabimo indukcijo po stopnji d. Za $d = 0$ trditev velja, saj smo že dokazali, da je $1 \in \mathcal{L}(\mathcal{B}^d)$. Naša indukcijska predpostavka je, da izrek velja za $d-1$. Za neka števila c_{ijk} torej velja

$$x^{m-1}y^n = \sum_{i+j+k=d-1} c_{ijk}B^{d-1}_{ijk}(x,y), \quad m+n \leq d, 1 \leq m.$$

Od tod sledi

$$\begin{aligned}
x^my^n &= x(x^{m-1}y^n) \\
&= (b_1x_1 + b_2x_2 + b_3x_3) \sum_{i+j+k=d-1} c_{ijk}B^{d-1}_{ijk}(x,y).
\end{aligned}$$

Uporabimo enako idejo kot pri zapisu monoma x zgoraj in dobimo

$$x^my^n = \sum_{i+j+k=d} d_{ijk}B^d_{ijk}(x,y)$$

za neka števila d_{ijk}.

Za neke konstante c_{ijk} in d_{ijk} očitno velja tudi

$$\begin{aligned} x^m y^n &= (x^m y^{n-1})y \\ &= \sum_{i+j+k=d-1} c_{ijk} B^{d-1}_{ijk}(x,y)(b_1y_1 + b_2y_2 + b_3y_3) \\ &= \sum_{i+j+k=d} d_{ijk} B^{d}_{ijk}(x,y). \end{aligned}$$

Poljuben bazni polinom prostora $\mathcal{P}_d$ smo zapisali v Bernsteinovi bazi in tako končali z dokazom izreka. □

Dokazali smo, da Bernsteinovi bazni polinomi tvorijo bazo prostora $\mathcal{P}_d$, od tod pa direktno sledi spodnja posledica.

Posledica 3.16. *Vsak polinom $p \in \mathcal{P}_d$ stopnje d se da zapisati kot*

$$p = \sum_{i+j+k=d} c_{ijk} B^d_{ijk}, \tag{3}$$

pri čemer so c_{ijk} ustrezne konstante, imenovane B-koeficienti polinoma p.

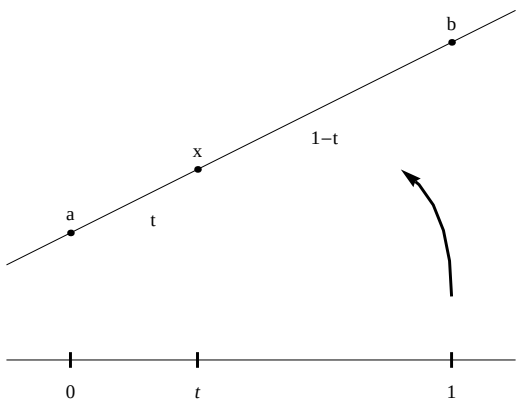
3.4. **De Casteljaujev algoritem za krivulje.** Ta algoritem velja za enega izmed najosnovnejših algoritmov, povezanih z oblikovanjem krivulj in ploskev. V pomoč nam bo pri geometrijski konstrukciji Bézierovih krivulj. Omenjeni algoritem namreč uporabimo za izračun vrednosti točk na Bézierovi krivulji.

Oglejmo si primer enostavne linearne interpolacije. Ta pojem potrebujemo v nalogi zato, ker je de Casteljaujev algoritem v geometrijskem smislu ponavljanje linearnih interpolacij.

3.4.1. *Linearna interpolacija.* Naj bosta a in b dve različni točki v $\mathbb{R}^3$. Premica skozi točki a in b je množica vseh točk $x \in \mathbb{R}^3$ oblike

$$x = x(t) = (1-t)a + tb, \quad t \in \mathbb{R}.$$

Na sliki 10 je grafični prikaz linearne interpolacije, kjer skozi točki a in b poteka premica. Točka x deli premico med a in b v razmerju $t : 1-t$.

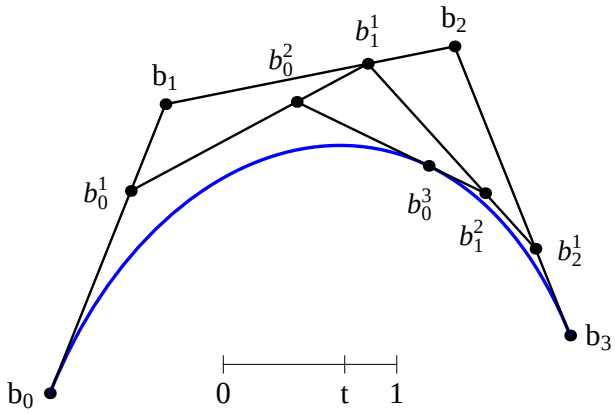


Slika 10. Linearna interpolacija

3.4.2. *Algoritem.* Podane imamo točke $b_j \in \mathbb{R}^3$ in $t \in \mathbb{R}$. De Casteljaujev algoritem izračuna

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t), \tag{4}$$

za $r = 1, 2, \ldots, n$ in $i = 0, 1, \ldots, n-r$. Velja, da je $b_i^0(t) = b_i$. Tedaj je $b_0^n(t)$ točka na Bézierovi krivulji b^n pri vrednosti parametra t.



Slika 11. De Casteljaujev algoritem za krivulje.

Izrek 3.17. *Vmesne de Casteljaujeve točke $b_i^r(t)$ lahko izrazimo z Bernsteinovimi polinomi stopnje r*

$$b_i^r(t) = \sum_{j=0}^{r} b_{i+j} B_j^r(t), \tag{5}$$

kjer je $r = 0, 1, \ldots, n$ in $i = 0, 1, \ldots, n-r$.

Dokaz. Izrek bomo dokazali s pomočjo indukcije. Za $r = 0$ dobimo

$$b_i^0(t) = b_i B_0^0(t) = b_i,$$

torej zgornja enačba velja. Indukcijska predpostavka pravi, da enačba (5) velja za $r-1$. Upoštevamo enačbo (4) in rekurzivno zvezo, ki velja za Bernsteinove polinome (2) ter izračunamo

$$
\begin{aligned}
b_i^r(t) &= (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t) \\
&= (1-t)\sum_{j=i}^{i+r-1} b_j B_{j-i}^{r-1}(t) + t\sum_{j=i+1}^{i+r} b_j B_{j-i-1}^{r-1}(t) \\
&= (1-t)\sum_{j=i}^{i+r} b_j B_{j-i}^{r-1}(t) + t\sum_{j=i}^{i+r} b_j B_{j-i-1}^{r-1}(t) \\
&= \sum_{j=i}^{i+r} b_j \left((1-t)B_{j-i}^{r-1}(t) + tB_{j-i-1}^{r-1}(t)\right) \\
&= \sum_{j=i}^{i+r} b_j B_{j-i}^{r}(t) \\
&= \sum_{j=0}^{r} b_{i+j} B_j^r(t)
\end{aligned}
$$

□

Glavni pomen enačbe (4) je v primeru, ko je $r = n$. Ustrezna de Casteljaujeva točka tedaj leži na krivulji

$$
b^n(t) = b_0^n(t) = \sum_{j=0}^{n} b_j B_j^n(t).
$$

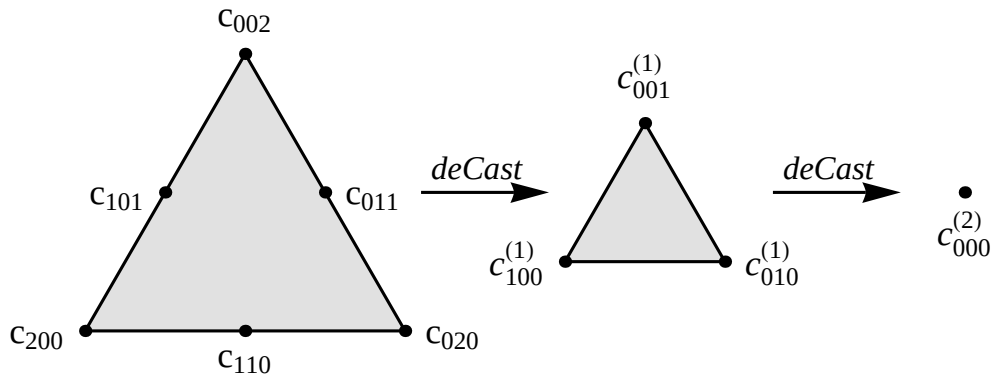
3.5. **De Casteljaujev algoritem za trikotne Bézierove krpe.** Pri tem gre za posplošitev osnovnega de Casteljaujevega algoritma za Bézierove krivulje, ki smo si ga ogledali v prejšnjem poglavju. Tudi tu gre za ponavljanje linearnih interpolacij.

Vhodni podatek algoritma je točka $v := (b_1, b_2, b_3)$ in koeficienti $\{c_{ijk}\}_{i+j+k=d}$. Želimo izračunati vrednost polinoma p v točki v tako, da določimo koeficient $c_{000}^{(d)}$.

Algoritem 1 De Casteljaujev algoritem

for $l = 1, 2, \ldots, d$ **do**
  for $i + j + k = d - l$ **do**
    $c_{ijk}^{(l)}(v) := b_1 c_{i+1,j,k}^{(l-1)(v)} + b_2 c_{i,j+1,k}^{(l-1)(v)} + b_3 c_{i,j,k+1}^{(l-1)(v)}$
return $c_{000}^{(d)}$

Predstavljen de Casteljaujev algoritem nam je v pomoč pri računanju vrednosti polinoma p v dani točki. Takšen izračun je namreč precej bolj ekonomičen kot pa direktno računanje vrednosti polinoma, zapisanega v Bernstein-Bézierovi bazi.



SLIKA 12. Zgled de Casteljaujevega algoritma za $d = 2$.

Izrek 3.18. *Dan je polinom p, zapisan v Bernstein-Bézierovi bazi kot v (3), in točka v z baricentričnimi koordinatami (b_1, b_2, b_3). Naj bodo koeficienti označeni s*

$$c_{ijk}^{(0)} := c_{ijk}, \quad i + j + k = d.$$

Za izračun koeficientov $c_{ijk}^{(l)}$ za $l = 1, 2, \ldots, d$ uporabimo de Casteljaujev algoritem in dobimo zvezo

$$c_{ijk}^{(l)} := b_1 c_{i+1,j,k}^{(l-1)} + b_2 c_{i,j+1,k}^{(l-1)} + b_3 c_{i,j,k+1}^{(l-1)}, \quad i + j + k = d - l. \tag{6}$$

Tedaj za $0 \leq l \leq d$ velja, da je

$$p(v) = \sum_{i+j+k=d-l} c_{ijk}^{(l)} B_{ijk}^{d-l}(v).$$

Posebej, za $l = d$ velja

$$p(v) = c_{000}^{(d)}.$$

Dokaz. Trditev najprej preverimo za $l = 0$. V tem primeru izrek očitno velja, saj dobimo ravno polinom, zapisan kot v (3). Nadaljujemo z indukcijo po l. Po indukcijski predpostavki izrek velja za $l - 1$, torej je

$$p(v) \quad = \sum_{i+j+k=d-l+1} c_{ijk}^{(l-1)} B_{ijk}^{d-l+1}(v).$$

Sedaj uporabimo že znan zapis Bernsteinovega polinoma iz definicije 3.13, ki ga prilagodimo potrebam dokaza. Pri pogoju $i + j + k = d$ dobimo

$$\begin{aligned} B_{ijk}^{d}(v) &= \frac{d!}{i!j!k!} b_1^i \, b_2^j \, b_3^k \\ &= \left(\frac{(d-1)!}{(i-1)!j!k!} + \frac{(d-1)!}{i!(j-1)!k!} + \frac{(d-1)!}{i!j!(k-1)!} \right) b_1^i \, b_2^j \, b_3^k \\ &= B_{i-1,j,k}^{d-1}(v) \cdot b_1 + B_{i,j-1,k}^{d-1}(v) \cdot b_2 + B_{i,j,k-1}^{d-1}(v) \cdot b_3. \end{aligned}$$

S pomočjo zveze, ki smo jo pravkar izpeljali, nadaljujemo z dokazom izreka. Vrednost polinoma p v točki v je torej

$$
\begin{aligned}
p(v) &= \sum_{i+j+k=d-l+1} c_{ijk}^{(l-1)} \left(b_1 B_{i-1,j,k}^{d-l}(v) + b_2 B_{i,j-1,k}^{d-l}(v) + b_3 B_{i,j,k-1}^{d-l}(v)\right) \\
&= \sum_{i+j+k=d-l+1} \left(c_{ijk}^{(l-1)} b_1 B_{i-1,j,k}^{d-l}(v) + c_{ijk}^{(l-1)} b_2 B_{i,j-1,k}^{d-l}(v) + c_{ijk}^{(l-1)} b_3 B_{i,j,k-1}^{d-l}(v)\right).
\end{aligned}
$$

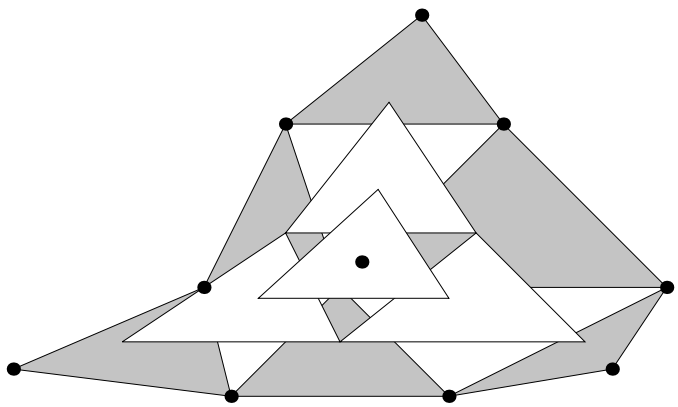
Ko zgornjo vsoto razdelimo na tri dele in zamaknemo indekse, dobimo zveze

$$
\begin{aligned}
\sum_{\substack{i+j+k=d-l+1 \\ i\geq 1}} c_{ijk}^{(l-1)} b_1 B_{i-1,j,k}^{d-l}(v) &= \sum_{i+j+k=d-l} c_{i+1,j,k}^{(l-1)} b_1 B_{ijk}^{d-l}(v), \\
\sum_{\substack{i+j+k=d-l+1 \\ j\geq 1}} c_{ijk}^{(l-1)} b_2 B_{i,j-1,k}^{d-l}(v) &= \sum_{i+j+k=d-l} c_{i,j+1,k}^{(l-1)} b_2 B_{ijk}^{d-l}(v), \\
\sum_{\substack{i+j+k=d-l+1 \\ k\geq 1}} c_{ijk}^{(l-1)} b_3 B_{i,j,k-1}^{d-l}(v) &= \sum_{i+j+k=d-l} c_{i,j,k+1}^{(l-1)} b_3 B_{ijk}^{d-l}(v).
\end{aligned}
$$

Seštejemo vse tri vsote in uporabimo predpostavko izreka (6). Dobimo

$$
\begin{aligned}
p(v) &= \sum_{i+j+k=d-l} \left(c_{i+1,j,k}^{(l-1)} b_1 + c_{i,j+1,k}^{(l-1)} b_2 + c_{i,j,k+1}^{(l-1)} b_3\right) B_{ijk}^{d-l}(v) \\
&= \sum_{i+j+k=d-l} c_{ijk}^{(l)} B_{ijk}^{d-l}(v).
\end{aligned}
$$

Za $l = d$ izrek očitno velja, saj je edini neničelni polinom B_{000}^0 stopnje 0 enak 1. $\square$



Slika 13. De Casteljaujev algoritem za trikotne Bézierove krpe.

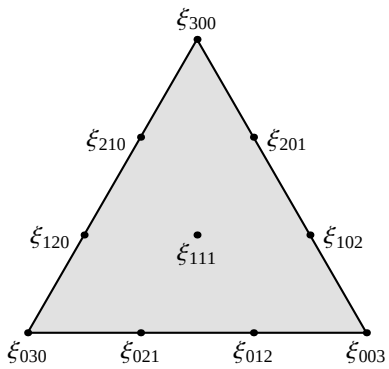
3.6. **Prostor zlepkov.** V podpoglavju 3.3 smo ugotovili, da je Bernstein-Bézierova baza precej bolj na mestu za uporabo nad trikotnikom kot pa standardna baza v prostoru polinomov $\mathcal{P}_d$. Dokazali smo tudi, da se da poljuben polinom zapisati v obliki (3). Posplošimo to na zlepke polinomov dveh spremenljivk.

Definicija 3.19. *Množico vseh točk domene trikotnika T z oglišči v točkah v_1, v_2 in v_3* označimo z

$$\mathcal{D}_{d,T} := \{\xi_{ijk}\}_{i+j+k=d},$$

pri čemer so točke domene definirane kot

$$\xi_{ijk} := \frac{iv_1 + jv_2 + kv_3}{d}, \quad i, j, k \in \mathbb{N}_0, \quad i + j + k = d.$$

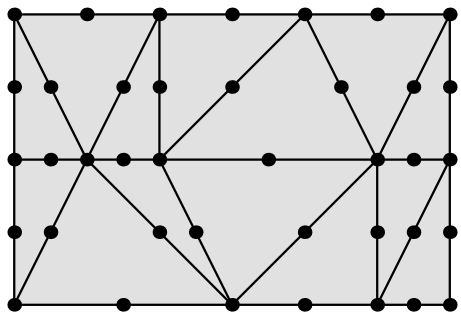


Slika 14. Točke domene za $d = 3$.

Definicija 3.20. Dana je regularna triangulacija $\Delta = \{T_1, T_2, \ldots, T_N\}$ območja Ω v ravnini. *Množica točk domene dane triangulacije* glede na d je

$$\mathcal{D}_{d,\Delta} := \bigcup_{T \in \Delta} \mathcal{D}_{d,T},$$

kjer je $\mathcal{D}_{d,T}$ množica točk domene trikotnika T.



Slika 15. Množica točk domene $\mathcal{D}_{2,\Delta}$ za dano triangulacijo Δ.

Definicija 3.21. Naj bo dana triangulacija $\Delta = \{T_1, T_2, \ldots, T_N\}$ območja Ω. *Vektorski prostor zveznih zlepkov polinomske dimenzije manjše ali enake d* je množica

$$\mathcal{S}_d^0(\Delta) := \{s \in \mathcal{C}^0(\Omega) : s|_{T_i} \in \mathcal{P}_d, \ i = 1, 2, \ldots, N\}.$$

Poljuben zlepek $s \in \mathcal{S}_d^0(\Delta)$ se da podobno kot v (3) zapisati v obliki

$$s|_T = \sum_{\xi \in \mathcal{D}_{d,T}} c_\xi B_\xi^{T,d},$$

kjer je T trikotnik triangulacije Δ, $\{c_\xi\}_{\xi \in \mathcal{D}_{d,T}}$ koeficienti in $B_\xi^{T,d}$ Bernsteinovi bazni polinomi stopnje d. To pomeni, da je zlepek $s \in \mathcal{S}_d^0(\Delta)$ enolično določen z množico koeficientov $\{c_\xi\}_{\xi \in \mathcal{D}_{d,\Delta}}$ in obratno. Ker zahtevamo, da je zlepek zvezen, se morajo koeficienti c_ξ na skupnem robu dveh trikotnikov ujemati.

3.7. **Pogoji gladkosti.** Želimo poiskati zlepek, ki bo zvezno odvedljiv. Zanima nas, kako ti vnaprej določeni pogoji gladkosti določajo vrednosti B-koeficientov.

Izrek 3.22. *Dana sta trikotnika* $T := \langle v_1, v_2, v_3 \rangle$ *in* $\widetilde{T} := \langle v_4, v_2, v_3 \rangle$ *s skupnim robom* $e := \langle v_2, v_3 \rangle$ *ter polinoma*

$$\begin{aligned} p(v) &:= \sum_{i+j+k=d} c_{ijk} B^d_{ijk}(v) \\ \widetilde{p}(v) &:= \sum_{i+j+k=d} \widetilde{c}_{ijk} \widetilde{B}^d_{ijk}(v) \end{aligned}$$

pri čemer sta $\{B^d_{ijk}\}$ *in* $\{\widetilde{B}^d_{ijk}\}$ *Bernsteinovi bazi glede na trikotnika* T *in* $\widetilde{T}$. *Zlepek, sestavljen iz polinomov* p *in* $\widetilde{p}$ *zadošča* C^r *gladkosti natanko takrat, ko je*

$$\widetilde{c}_{njk} = \sum_{i'+j'+k'=n} c_{i',j'+k,k'+j} B^n_{i'j'k'}(v_4) \tag{7}$$

za vse $j + k = d - n$ *in* $n = 0, 1, \ldots, r$.

Dokaz najdemo v [9].

Ker nas zanima zlepek, ki bo zvezno odvedljiv, si bomo ogledali geometrijski primer $\mathcal{C}^1$ gladkosti.

Še vedno naj veljajo vse predpostavke izreka 3.22 in naj bo točka v_4 zapisana v baricentričnih koordinatah glede na trikotnik T kot $v_4 := (b_1, b_2, b_3)$. Kontrolne točke polinomov p in $\widetilde{p}$ definiramo kot točke $C_{ijk} := (\xi_{ijk}, c_{ijk}) \in \mathbb{R}^3$ in $\widetilde{C}_{ijk} := (\widetilde{\xi}_{ijk}, \widetilde{c}_{ijk}) \in \mathbb{R}^3$. Najprej preverimo pogoj za zveznost ($r = 0$). Iz enačbe (7) dobimo

$$\widetilde{c}_{0jk} = c_{0kj}, \quad j + k = d.$$

To pomeni, da sta polinoma p in $\widetilde{p}$ na robu e zvezno povezana natanko tedaj, ko se ujemajo njune kontrolne točke nad robom e, kot vidimo na sliki 16.

Pogoj, ki nam zagotavlja zvezno odvedljivost ($r = 1$) pa določa, da je

$$\widetilde{c}_{1,j,k} = b_1 c_{1,k,j} + b_2 c_{0,k+1,j} + b_3 c_{0,k,j+1}, \quad j + k = d - 1. \tag{8}$$

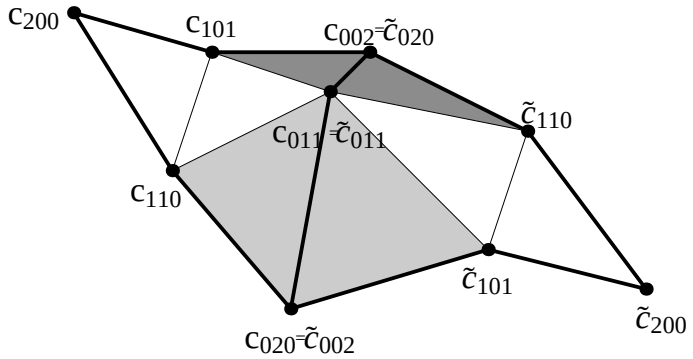
Ker sta si trikotnika $\langle v_1, v_2, v_3 \rangle$ in $\langle \xi_{i+1,j,k}, \xi_{i,j+1,k}, \xi_{i,j,k+1} \rangle$ podobna, velja

$$\xi_{1,j,k} = b_1 \xi_{1,k,j} + b_2 \xi_{0,k+1,j} + b_3 \xi_{0,k,j+1}, \quad j + k = d - 1. \tag{9}$$

Z združitvijo zgornjih enačb dobimo zvezo, ki velja za vse $j + k = d - 1$:

$$\widetilde{C}_{1,j,k} = b_1 C_{1,k,j} + b_2 C_{0,k+1,j} + b_3 C_{0,k,j+1}.$$

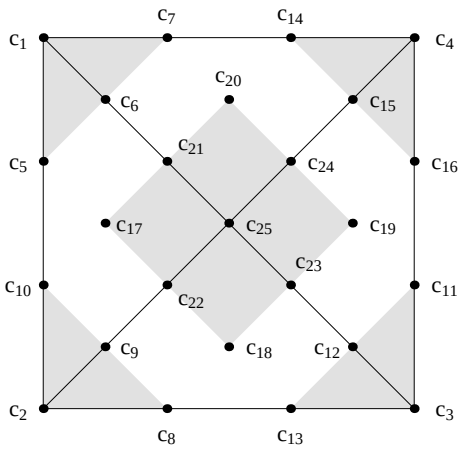
To pomeni, da mora biti za zadostitev pogoja $\mathcal{C}^1$ gladkosti vsaka kontrolna točka $\widetilde{C}_{1,j,k}$ linearna kombinacija treh točk iz sosednjega trikotnika. Omenjene štiri točke torej ležijo na isti ravnini (glej sliko 16).



Slika 16. Geometrijska interpretacija $\mathcal{C}^1$ gladkosti pri $d = 2$.

3.8. **Prostor $\mathcal{C}^1$ štirikotnih makroelementov.** V poglavju 7 smo za reševanje problema uporabili posebno vrsto makroelementov, štirikotni makroelement. Ker podrobnosti o temi presegajo področje moje diplomske naloge, si bomo ogledali le konkreten primer in konstrukcijo omenjenega makroelementa.

Dana je pravokotna mreža točk, ki jo imenujemo *kvadrangulacija* in jo označimo z $\square$. Spomnimo se triangulacije tipa 2 (slika 8), ki smo jo dobili tako, da smo vsakemu izmed pravokotnikov dodali obe diagonali. Na enak način sedaj kvadrangulacijo $\square$ z dodajanjem obeh diagonal pretvorimo v triangulacijo Δ. Vsakega izmed pravokotnikov smo torej razdelili na štiri trikotnike. Oglejmo si primer za en makroelement $Q := \langle v_1, v_2, v_3, v_4 \rangle$. Koeficiente pripadajočih točk domene označimo analogno, kot vidimo na sliki 17.



Slika 17. Koeficienti posameznega makroelementa.

Konstruirati želimo zvezno odvedljiv kubični zlepek $\mathcal{S}_3^1(\Delta)$. Koeficienti c_1, c_2, c_3 in c_4 na robu makroelementa s so zaradi pogojev zveznosti natančno določeni z višinami danih točk

$$c_i = s(v_i), \quad i = 1, 2, 3, 4.$$

Za vsak rob makroelementa $e := \langle u, v \rangle$ poiščemo točko $w_e := (u + v)/2$ na sredini roba e. V točki w_e določimo smerni odvod D_{u_e} v smeri enotskega vektorja u_e, ki ga dobimo z vrtenjem roba e za 90 stopinj v nasprotni smeri urinega kazalca. Točka

$v_Q := (x_Q, y_Q)$ leži na presečišču diagonal makroelementa Q.

Naslednjih dvanajst koeficientov zlepka s določimo s pomočjo formul:

$$\begin{aligned}
c_5 &= [(x_2 - x_1)s_x(v_1) + (y_2 - y_1)s_y(v_1)]/3 + s(v_1) \\
c_6 &= [(x_Q - x_1)s_x(v_1) + (y_Q - y_1)s_y(v_1)]/3 + s(v_1) \\
c_7 &= [(x_4 - x_1)s_x(v_1) + (y_4 - y_1)s_y(v_1)]/3 + s(v_1) \\
c_8 &= [(x_3 - x_2)s_x(v_2) + (y_3 - y_2)s_y(v_2)]/3 + s(v_2) \\
c_9 &= [(x_Q - x_2)s_x(v_2) + (y_Q - y_2)s_y(v_2)]/3 + s(v_2) \\
c_{10} &= [(x_1 - x_2)s_x(v_2) + (y_1 - y_2)s_y(v_2)]/3 + s(v_2) \\
c_{11} &= [(x_4 - x_3)s_x(v_3) + (y_4 - y_3)s_y(v_3)]/3 + s(v_3) \\
c_{12} &= [(x_Q - x_3)s_x(v_3) + (y_Q - y_3)s_y(v_3)]/3 + s(v_3) \\
c_{13} &= [(x_2 - x_3)s_x(v_3) + (y_2 - y_3)s_y(v_3)]/3 + s(v_3) \\
c_{14} &= [(x_1 - x_4)s_x(v_4) + (y_1 - y_4)s_y(v_4)]/3 + s(v_4) \\
c_{15} &= [(x_Q - x_4)s_x(v_4) + (y_Q - y_4)s_y(v_4)]/3 + s(v_4) \\
c_{16} &= [(x_3 - x_4)s_x(v_4) + (y_3 - y_4)s_y(v_4)]/3 + s(v_4).
\end{aligned}$$

Pogoji zvezne odvedljivosti določajo še preostale koeficiente. Koeficienti so izbrani tako, da interpoliramo točke in prve parcialne odvode v ogliščih pravokotnikov, na sredini povezav pa interpoliramo smerne odvode. Podrobnejšo razlago najdemo v [9].

Opazimo, da so za vsak zlepek $s \in \mathcal{S}_1^3(\Delta)$ in trikotnik $T \in \Delta$ koeficienti polinoma $s|_T$ izraženi s podatki znotraj trikotnika T. To pomeni, da dobimo iz interpolacijskih podatkov enolično določen zlepek, ki je odvisen le od lokalnih podatkov.

4. Kriterij optimalnosti in energijski funkcional

Optimalna pot, ki jo bomo dobili kot rezultat predstavljenega algoritma, bo odvisna od izbire kriterija optimalnosti. V tej nalogi smo se odločili za minimalno porabo energije, potrebne za vzpon na goro, brez težav pa bi lahko določili drugačen kriterij. Možnosti je več, od iskanja najkrajše poti do najkrajšega časa vzpona, možno pa je izbrati celo kombinacijo več že omenjenih kriterijev.

4.1. Energijski funkcional.

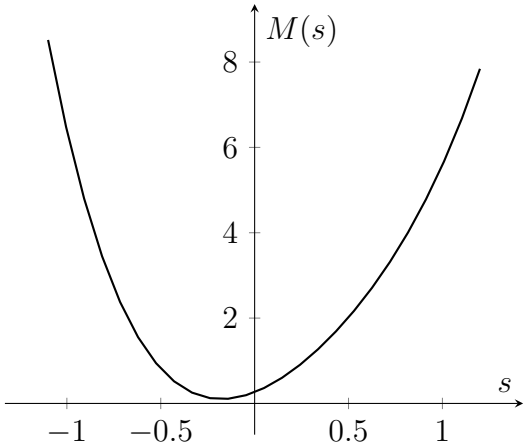
Naš cilj bo poiskati optimalno pot, ki bo potekala od izbrane začetne točke (izhodišča planinske poti) pa do končne točke oziroma vrha gore, glede na predhodno izbran kriterij. Ker želimo izračunati porabo energije pri premiku vzdolž vsake izmed robnih krivulj makroelementa, najprej potrebujemo funkcijo, ki podaja količino porabljene energije na enoto prehojene poti.

S pomočjo empiričnih poskusov na posameznikih in aproksimacijo meritev so v [10] po metodi najmanjših kvadratov prišli do polinoma četrte stopnje M, ki podaja porabo energije na enoto prehojene poti za povprečnega pohodnika.

Definicija 4.1. Funkcija

$$M(s) = 0.2635 + 1.737\,s + 4.237\,s^2 - 2.143\,s^3 + 1.493\,s^4$$

predstavlja porabo energije na enotsko dolžino poti za povprečnega pohodnika. Pri tem je s tangens naklonskega kota terena v smeri gibanja, enota za funkcijo M pa so kJ/m.



Slika 18. Graf porabe energije (kJ/m) na meter prehojene poti v odvisnosti od naklona.

Očitno je, da je funkcija M odvisna od naklona terena v smeri hoje, seveda pa velja tudi dejstvo, da ni simetrična glede na gibanje po klancu navzgor ali navzdol. Graf funkcije M na sliki 18 potrdi naše domneve iz prakse, da človek najmanj energije porabi za gibanje po rahlem klancu navzdol. Omenjeno je lepo vidno na sliki 18, kjer negativni del predstavlja spust, pozitivni del abscisne osi pa vzpon.

Iz izkušenj pa nam je znano še eno dejstvo in sicer, da se pri nekem določenem naklonu terena začnemo gibati v t. i. cikcak vzorcu. Na ta način premagujemo velike strmine terena in kljub daljši prehojeni poti prihranimo pri celotni porabi energije. Z analizo poti pri konstantnem naklonu terena in uporabo energijskega funkcionala

$$I(\delta) := \frac{M(\tan\theta\cos\delta)}{\cos\delta},$$

so v [10] naše opažanje iz vsakdanjega življenja potrdili. Pri tem je s θ označen naklonski kot terena, δ pa predstavlja kot med trenutno smerjo gibanja in smerjo, ki povezuje točko, kjer se pohodnik nahaja v danem trenutku in točko, ki predstavlja vrh gore. V tem primeru je naklon terena v trenutni smeri enak $s = \tan\theta\cos\delta$. Kritičen kot, ki so ga na ta način določili, znaša za vzpenjanje na goro 15.6°, za hojo po klancu navzdol pa 12.5°.

Ker gre pri temi diplomskega seminarja za proučevanje optimalnega vzpona na goro, moramo zgornjo definicijo energijskega funkcionala preko integrala razširiti za primer nekonstantnega naklona.

Naj bo $\boldsymbol{r} : [0,1] \to \mathbb{R}^3$ robna krivulja makroelementa, ki smo ga predstavili v prejšnjih poglavjih. Pri tem je $\boldsymbol{r} = (r_x, r_y, r_z)^T$ vektorska funkcija parametra $t \in [0,1]$. Sedaj že vemo, da lahko krivuljo $\boldsymbol{r}$ zapišemo v Bézierovi obliki

$$\boldsymbol{r} = \sum_{j=0}^{d} c_j B_j^d.$$

Njen odvod je tedaj enak

$$\dot{\boldsymbol{r}} = (\dot{r}_x, \dot{r}_y, \dot{r}_z)^T = \sum_{j=0}^{d-1} d(c_{j+1} - c_j) B_j^{d-1}.$$

Zaradi nekonstantnega naklona vzdolž krivulje uporabimo infinitezimalne premike in tako dobimo prispevek energije E na delu loka krivulje dl

$$dE(\boldsymbol{r}) = M(s(t))dl,$$

pri čemer je

$$s(t) = \tan\varphi(t) = \frac{\dot{r}_z(t)}{\sqrt{\dot{r}_x^2(t) + \dot{r}_y^2(t)}}.$$

Z upoštevanjem

$$dl = ||\dot{\boldsymbol{r}}(t)|| = \sqrt{\dot{r}_x^2(t) + \dot{r}_y^2(t) + \dot{r}_z^2(t)}\, dt,$$

dobimo formulo za porabo energije na celotni krivulji:

$$E(\boldsymbol{r}) = \int_0^1 M(s(t))||\dot{\boldsymbol{r}}(t)||dt.$$

5. Optimizacijski problem

Za konec želimo dobiti pot, ki bo rešila naš optimizacijski problem. To je tista pot, ki bi jo izbrali, če bi želeli pri vzponu na goro porabiti čim manj energije. Energijski funkcional E iz prejšnjega razdelka bomo uporabili na vsaki izmed robnih krivulj makroelementa in tako celotno nalogo iskanja optimalne poti prevedli na iskanje najcenejše poti v uteženem grafu. Reševanja tega problema se bomo lotili s pomočjo *Dijkstrovega algoritma*.

5.1. **Dijkstrov algoritem.** Omenjeni algoritem, imenovan po nizozemskem matematiku in fiziku E. W. Dijkstri, je orodje za iskanje najcenejše poti v usmerjenem grafu $G = (V, E)$. Množica točk je označena z V, E pa predstavlja množico uteženih povezav z nenegativno utežjo oz. ceno $w_e : E \to [0, \infty)$. Cena povezave ima lahko v različnih problemih različen pomen. Označuje lahko ceno prevoza po poti od enega krajišča povezave do drugega, dolžino poti med točkama ali pa, denimo, kot v našem primeru, porabo energije na povezavi, ki predstavlja delček prehojene poti.

Oglejmo si delovanje Dijkstrovega algoritma v splošnem. Naj bo dan usmerjen graf $G = (V, E)$, množica točk V, množica povezav E s cenami $w_e \geq 0$ in začetna točka s. S P označimo neko pot od ene točke v grafu do druge. To pomeni, da je P množica vseh povezav na tej poti. Cena poti P bo torej vsota cen vseh povezav, ki so v P.

Naj bo S množica tistih točk $u \in V$, za katere že poznamo najcenejšo pot $c(u)$ od točke s. Tej množici bomo rekli *raziskan del grafa*. Očitno je na začetku v množici S le točka s, torej $S = \{s\}$ in $c(s) = 0$. Nadalje algoritem deluje tako, da za vsako točko iz neraziskanega dela grafa $v \in V - S$ določi najcenejšo pot od s do v. Iskanje poteka s potovanjem po raziskanem delu grafa oz. množici S od točke s do neke točke $u \in S$ in nato naprej po povezavi od u do v. Zanima nas cena poti $c(u) + w_e$,

pri čemer je z e označena povezava od u do v. Ker iščemo najcenejšo pot, opazujemo količino

$$c := \min_{\substack{e=uv:\\ u\in S}} (c(u) + w_e),$$

nato pa izberemo tisto točko $v \in V - S$, za katero zgornja količina doseže minimum. Izbrano točko v dodamo v raziskan del grafa oz. množico S in definiramo $c(v) = c$. Na vsakem koraku shranimo povezavo uv z minimalno ceno, saj bodo na koncu te povezave tvorile najcenejšo pot, ki jo iščemo. Najcenejšo pot P_v od s do v dobimo tako, da se po shranjenih povezavah premikamo od točke v pa vse do točke s.

V našem primeru smo Dijkstrov algoritem za iskanje optimalne poti na goro pognali v Mathematici. Vhodni podatek je utežen graf $G = (V, E)$, ki je v našem primeru kar triangulacija na točkah v_i. V množici točk V so oglišča trikotnikov, množico povezav E pa tvorijo njihove stranice. Če z $\boldsymbol{r}$ označimo robno krivuljo makroelementa nad stranico, ki ustreza povezavi e, potem določimo ceno povezave e tako, da je

$$w_e := E(\boldsymbol{r}),$$

pri čemer je E energijski funkcional, izpeljan v prejšnjem poglavju.

6. Opis problema in predstavitev uporabljenih algoritmov

V tem poglavju bomo v strnjeni obliki zapisali osnovne korake in orodja, ki jih bomo potrebovali za iskanje optimalne poti in izračun količine porabljene energije pri vzponu na goro. Za reševanje tega težkega problema je potrebno predhodno poglobljeno znanje iz numerične analize, optimizacije in diskretne matematike. Nekatere teme smo v prejšnjih razdelkih pogledali bolj podrobno, drugih pa smo se z razlago le dotaknili in smo opisali zgolj ideje. Implementacija celotnega algoritma je zapisana v Mathematici [13].

Vhodni podatki omenjenega algoritma so točke na terenu in začetna ter končna točka, ki predstavljata izhodišče planinske poti in vrh gore. Izhodna podatka bosta optimalna pot, ki jo iščemo, in poraba energije E vzdolž te poti.

Za preizkus delovanja algoritma moramo najprej pridobiti GPS podatke terena. Pravokotna mreža točk, ki jo na ta način dobimo, je običajno prevelika, saj zajema še velik del okolice izbrane gore, ki pa za izračune ni relevanten. Zato mrežo točk najprej obrežemo in jo razredčimo, da dobimo manjše število podatkov, ki pa še vedno zadostuje za dovolj natančne izračune.

Nadalje določimo kvadrangulacijo, razbitje naše domene na pravokotnike. Na ta način dobimo seznam točk, ki predstavljajo oglišča pravokotnikov. Vsakega izmed pravokotnikov kvadrangulacije razdelimo na štiri manjše trikotnike (triangulacija), kot smo opisali v poglavju 3.

Parcialne odvode prvega reda, ki jih potrebujemo za določitev koeficientov makroelementa, aproksimiramo s končnimi diferencami danih točk.

Sledi metoda, ki določi kontrolne koeficiente (B-koeficiente) makroelementa za izbrano triangulacijo. Za risanje slike reliefa in poti na njem potrebujemo metodo, ki zgenerira sliko nad dano triangulacijo. Pri tem zaradi njihovih ugodnejših lastnosti uporabimo baricentrične koordinate in de Casteljaujev algoritem.

Prostor iskanja poti diskretiziramo in optimalno pot poiščemo na danem uteženem grafu oz. mreži poti. Uteži oz. cene povezav predstavljajo porabo energije pri vzponu na goro. Izračunane so s pomočjo kriterijske (energijske) funkcije na vsakem izmed robov makroelementa, ki smo jih predstavili v obliki Bézierovih krivulj. Na ta način lahko brez težav dobimo porabo energije na celotni poti. Optimalno pot določimo s pomočjo Dijkstrovega algoritma.

7. Dva primera izračuna optimalnih poti

Po predstavitvi algoritma za iskanje optimalnih poti in utemeljitvi teorije, ki leži v ozadju tega problema, je njegovo učinkovitost najbolje preveriti na konkretnem primeru. V tem poglavju si bomo najprej ogledali rezultate, dobljene pri iskanju optimalne poti na Šmarno goro, predstavljene v [5] in [6]. Nato bomo algoritem testirali še na primeru Triglava. Celotna implementacija je narejena s programskim paketom Mathematica [13].

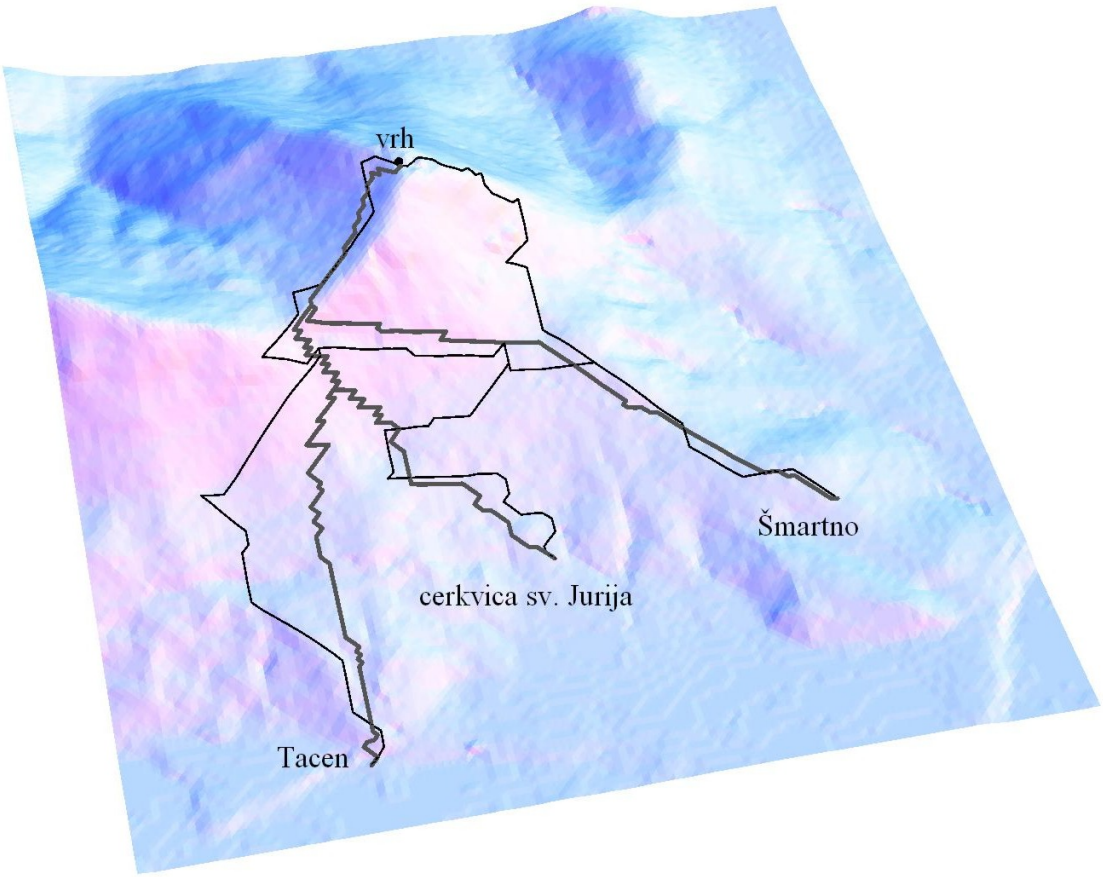
7.1. **Šmarna gora.** Prvi zgled bo 669 m visok vrh severozahodno od Ljubljane, Šmarna gora. Vzeli bomo štiri priljubljene pohodniške poti s tremi različnimi izhodišči. Primerjali bomo porabo energije na poteh, kot potekajo v naravi, in izračunanih optimalnih poteh glede na različna izhodišča in vrh.

Optimalne poti so bile v [5] izračunane na podlagi približno 40.000 podatkov, ki opisujejo relief Šmarne gore. Uporabljena je bila triangulacija tipa 2 (slika 8) kar pomeni, da so vsakega izmed pravokotnikov kvadrangulacije razdelili na štiri manjše trikotnike tako, da so dodali obe diagonali. Nad tako dobljeno triangulacijo so konstruirali kubični C^1 interpolacijski zlepek in na ta način dobili rezultate, predstavljene v tabeli 1. Spodnja tabela prikazuje porabo energije pri vzponu na Šmarno goro po različnih poteh.

Za lažjo predstavo nam služi grafična predstavitev rezultatov. Na sliki 19 so naravne poti narisane s tanko črno črto, izračunane optimalne poti pa so odebeljene. Kot vidimo, se optimalne poti precej dobro ujemajo z dejanskimi potmi. Največja odstopanja se pojavijo na strmejših predelih, kjer optimalna pot oblikuje cikcak vzorec.

Izhodišče	Pot	Poraba (kJ)	Prihranek (kJ)	Razlika (%)
Tacen (čez Sp. kuhinjo)	naravna	1589	186	11,7
	optimalna	1403		
cerkev Sv. Jurija (Romarska pot)	naravna	1484	232	15,6
	optimalna	1252		
Šmartno (Partizanska pot)	naravana	1456	206	14,1
	optimalna	1250		
Šmartno (Šmarska pot)	naravna	1403	153	10,9
	optimalna	1250		

Tabela 1. Poraba energije pri vzponu na Šmarno goro po različnih poteh.



Slika 19. Naravne poti (tanke črte) in izračunane optimalne poti (odebeljene črte) na Šmarno goro.

Primerjavo naravnih in optimalnih poti na sliki 19 si oglejmo nekoliko podrobneje. Opazimo, da se prva izmed poti, pot iz Tacna čez Spodnjo kuhinjo, kmalu odcepi od optimalne in naredi ovinek po zahodni strani vzpetine. Na ta način se namreč pot, ki je primerna za vse pohodnike in je označena kot lahka, izogne prestrmemu in prezahtevnemu direktnemu vzponu, ki ga kot rezultat vrne naš algoritem.

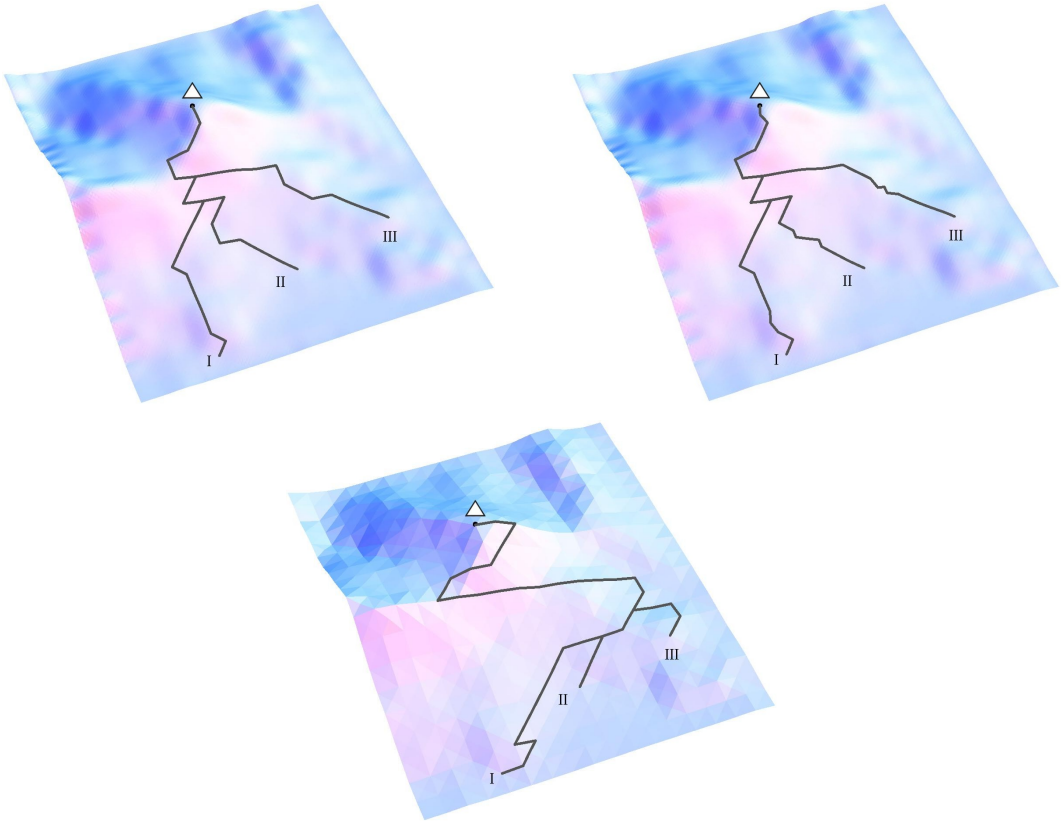
Romarska pot izpred cerkve Sv. Jurija je precej daljša od svoje različice, optimalne poti, in je zato glede na porabo energije napornejša. Ta pot obide vse strmejše predele, ki jih je težje prečkati in je zato vseskozi bolj položna od izračunane. Ta pot je v naravi namreč primerna tudi za manj pripravljene pohodnike in starejše ljudi.

Zadnji dve poti imata obe izhodišče v Šmartnem. Prva, t. i. Partizanska pot, je bila v drugi svetovni vojni v uporabi s strani kurirjev, zato se po pričakovanjih ves čas strmo vzpenja proti vrhu. To je pot, ki bi jo dobili, če bi za kriterij optimalnosti namesto minimalne porabe energije vzeli najkrajši možni čas.

Zadnja izmed poti, ki smo jih vzeli za primerjavo, je Šmarska pot. Opazimo, da je to pot, ki se najbolje prilega optimalni in pri kateri so odstopanja najmanjša. Tudi odstotek energije, ki jo prihranimo pri vzponu po njej (glej tabelo 1), je najmanjši izmed vseh štirih omenjenih poti.

Kot smo povedali že v prejšnjih poglavjih, so postopki iskanja optimalne poti in s tem dobljeni rezultati različni glede na obliko interpolacijske ploskve. Ker je lahko pri velikem številu podatkov časovna zahtevnost algoritmov precej velika, se nam porodi vprašanje, koliko podatkov v resnici potrebujemo za dovolj dobro aproksimacijo.

V primeru Šmarne gore se je izkazalo, da dobimo dobre približke poti že z uporabo precej manj podatkov. Na sliki 20 (zgoraj levo) vidimo, da se optimalne poti, pridobljene z le 900 enakomerno razredčenimi GPS točkami dobro ujemajo s potmi s slike 19, pridobljene z nekaj več kot 40.000 podatki. Po enem koraku subdivizije so rezultati še bolj natančni (slika 20 desno), medtem ko je z uporabo odsekoma linearne ploskve (slika 20 spodaj) aproksimacija reliefa preslaba.



Slika 20. Optimalne poti na Šmarno goro na grobi mreži podatkov (zgoraj levo), po enem koraku subdivizije (zgoraj desno) in na odsekoma linearni ploskvi (spodaj).

Da bo primerjava še bolj jasna, si oglejmo tabelo 2. V njej je zapisana poraba energije pri vzponu na Šmarno goro po izračunanih optimalnih poteh s treh različnih izhodišč, glede na obliko interpolacijske ploskve. Opazimo, da so poti nad grobo mrežo podatkov le malo slabše od tistih, ki smo jih dobili po enem koraku subdivizije. Odstopanja pri odsekoma linearni ploskvi so precej večja, znašajo tudi do štirideset odstotkov.

Izhodišče	Groba mreža	Po enem koraku subdivizije	Odsekoma linearna ploskev
Tacen	1373	1371	1935
cerkev Sv. Jurija	1281	1279	1699
Šmartno	1194	1191	1632

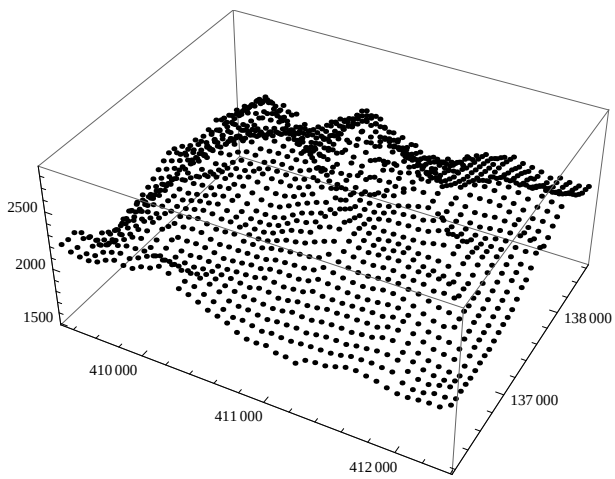
TABELA 2. Poraba energije (v kJ) pri vzponu na Šmarno goro glede na vrsto interpolacijske ploskve.

7.2. **Triglav.** Celoten postopek iskanja optimalne poti na goro smo spoznali in proučili na podlagi primera Šmarne gore iz [5] in [6]. Ker smo se želeli s problemom seznaniti še podrobneje, smo se odločili, da delovanje in učinkovitost omenjenega algoritma preverimo na novem primeru. V kopici možnosti, ki se nam ponujajo v Sloveniji, se nam je zdela najprimernejša in najzanimivejša naša najvišja gora, Triglav.

Iz baze Geodetske uprave Republike Slovenije smo dobili podatke digitalnih modelov višine, izdane v formatu XYZ:

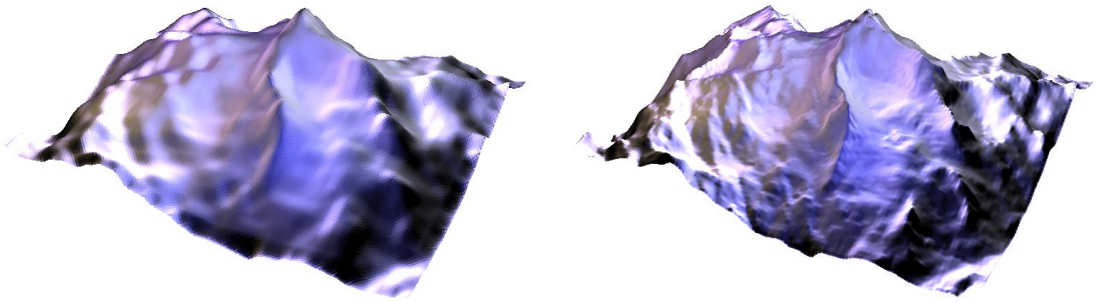
$$\begin{array}{lll} 407750{,}0 & 136000{,}0 & 2151{,}70 \\ 407755{,}0 & 136000{,}0 & 2150{,}30 \\ 407760{,}0 & 136000{,}0 & 2149{,}20 \\ & \vdots & \end{array}$$

Na ta način smo dobili pravokotno mrežo točk velikosti 6750 m $\times$ 2600 m na širšem območju Triglava. Z metodo za rezanje in redčenje smo ustvarili redkejšo mrežo višinskih točk (slika 21), ki pa nam kljub manjšemu številu podatkov še vedno da precej dober obris reliefa Triglava.

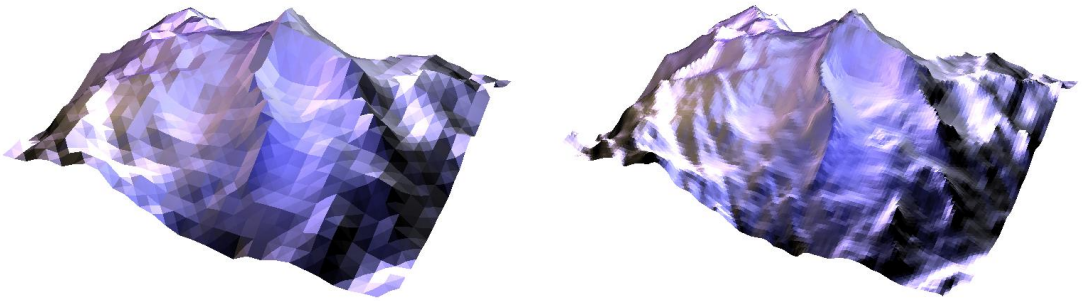


SLIKA 21. Višinske točke Triglava.

Preizkušali smo, kako se na obliki reliefa odraža število interpolacijskih podatkov in vrsta interpolacijske ploskve. Na slikah 22 in 23 lahko primerjamo izgled reliefa pri različnem številu interpolacijskih podatkov. Slika 22 prikazuje kubični $\mathcal{C}^1$ zlepek, ki smo ga konstruirali iz skoraj 12000 podatkov (desno) oz. le nekaj manj kot 800 podatkov (levo). Na enako gostih mrežah točk smo konstruirali še linearni zlepek in dobili rezultate na sliki 23.



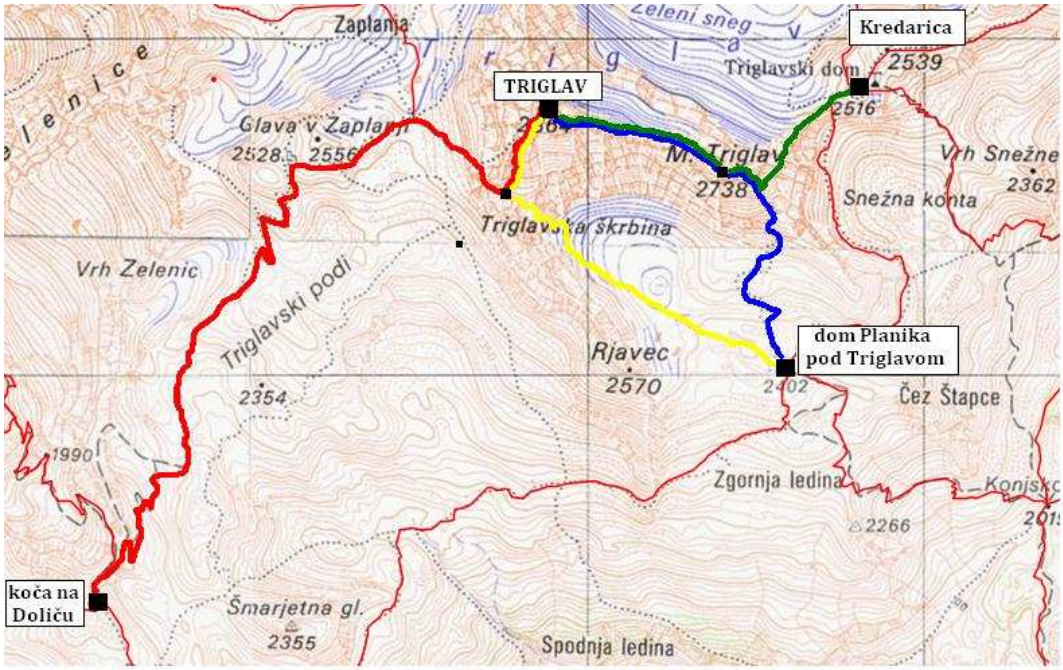
Slika 22. Primer kubičnega $\mathcal{C}^1$ zlepka na grobi mreži (levo) in finejši mreži (desno).



Slika 23. Primer odsekoma linearne ploskve na grobi mreži (levo) in finejši mreži (desno).

Z zgornjih slik je jasno vidno dejstvo, ki smo ga omenili že na začetku te naloge. Gladek zlepek je za opis reliefa bolj primeren od odsekoma linerne ploskve, saj se precej bolj približa naravnemu izgledu.

Po predstavitvi reliefa smo z uporabo postopkov, predstavljenih v prejšnjih poglavjih in prilagojenemu algoritmu iz [5], želeli dobiti optimalne poti in njihovo primerjavo z naravnimi potmi. Ker je problem Triglava zaradi njegove višine in strmih poti kompleksnejši, za izhodišča poti nismo vzeli točk v dolini, ampak koče, ki predstavljajo postojanke na celotni poti na Triglav.



Slika 24. Zemljevid območja Triglava z vrisanimi potmi.

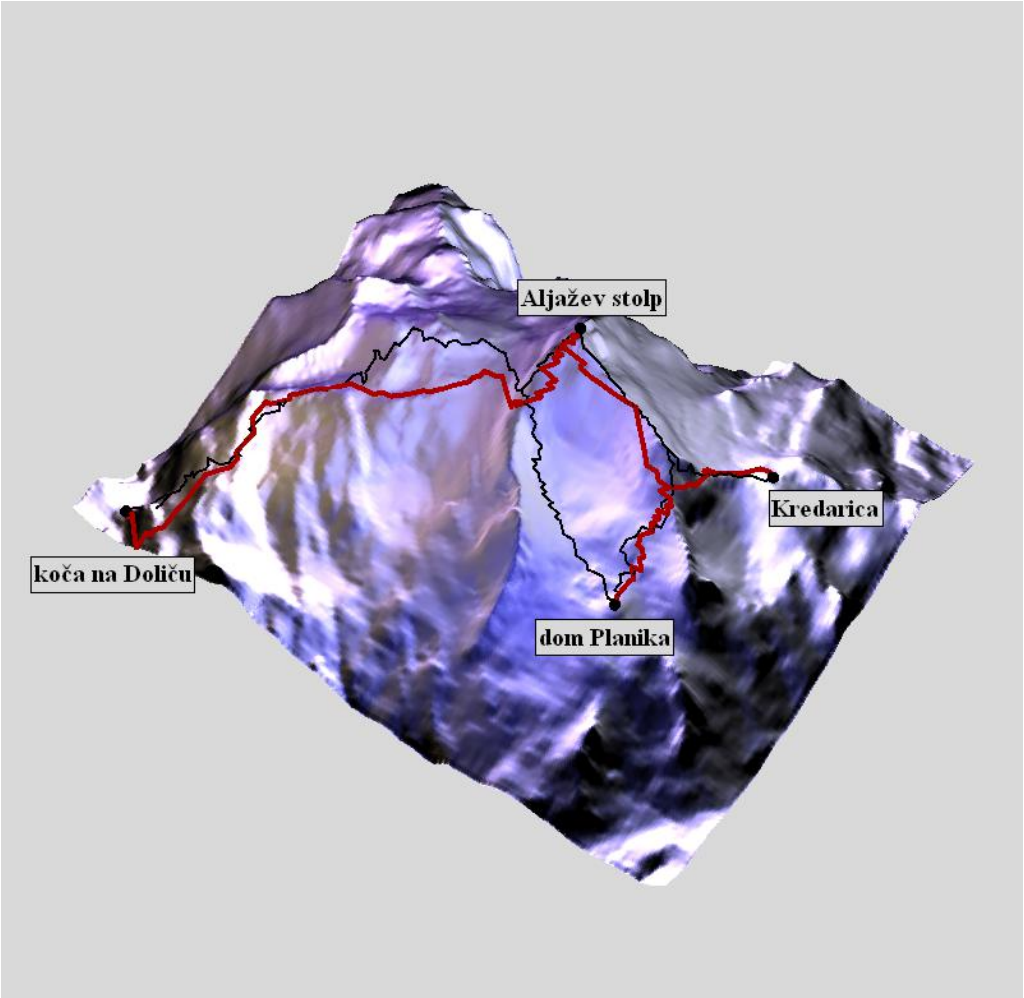
Prva izmed izbranih poti je pot, ki je na sliki 24 označena z rumeno barvo. Njeno izhodišče je dom Planika pod Triglavom (2401 m), vodi pa po strmi poti čez Triglavsko škrbino. V naravi poteka po meliščih in je zaradi izpostavljenosti padajočemu kamenju ter vzpenjanju s pomočjo klinov in jeklenic označena kot zelo zahtevna. Na sliki 25 vidimo, da se optimalna pot, ki vodi od doma Planika pa do vrha Triglava, odcepi od naravne že takoj na začetku in se na ta način izogne težko prehodnemu terenu. Optimalna pot ves čas poteka precej podobno drugi poti z istim izhodiščem, ki vodi čez Mali Triglav (na sliki 24 obarvana z modro). Ta pot je v naravi občasno precej strma, vendar dobro varovana, del poti poteka direktno po grebenu. Opažanja potrdijo podatki o porabi energije, zbrani v tabeli 3. Prihranek energije pri prvi poti je približno 56-odstoten, medtem ko se pri drugi poti, ki poteka podobno optimalni, zmanjša za več kot tretjino.

Naslednja pot se s Kredarice (2515 m) sprva zmerno spušča proti prevalu med Kredarico in Malim Triglavom, kmalu pa se prične vzpenjati po razčlenjenem skalovju. Nato sledi plezalni del. Pot je tehnično zelo zahtevna, saj se prečno vzpne po izpostavljeni polici, ki pa je sicer varovana s klini. Nato se priključi prvi poti in od Malega Triglava dalje poteka po enaki trasi. Optimalna pot v izogib plezalnemu delu, kjer porabimo veliko energije, poteka levo od naravne. Odstotek prihranka energije pri vzponu po optimalni poti s Kredarice je največji izmed vseh štirih poti.

Zadnja pot, ki smo jo izbrali za primerjavo, se prične na koči na Doliču (2151 m). Poteka po mulatjeri in se nato strmo vzpne po melišču, vse do stene s številnimi varovali. Optimalna pot od dejanske ne odstopa preveč, velik ovinek po desni strani naredi le na najbolj strmem področju (glej sliko 25).

Izhodišče	Pot	Poraba (kJ)	Prihranek (kJ)	Razlika (%)
dom Planika (čez Triglavsko škrbino)	naravna optimalna	4677,78 2055,10	2622,68	56,07
dom Planika (čez Mali Triglav)	naravna optimalna	3154,53 2055,10	1099,43	34,85
Kredarica (čez Mali Triglav)	naravna optimalna	4682,56 1723,50	2959,06	63,19
koča na Doliču (čez Triglavsko škrbino)	naravna optimalna	4950,75 3115,47	1835,28	37,07

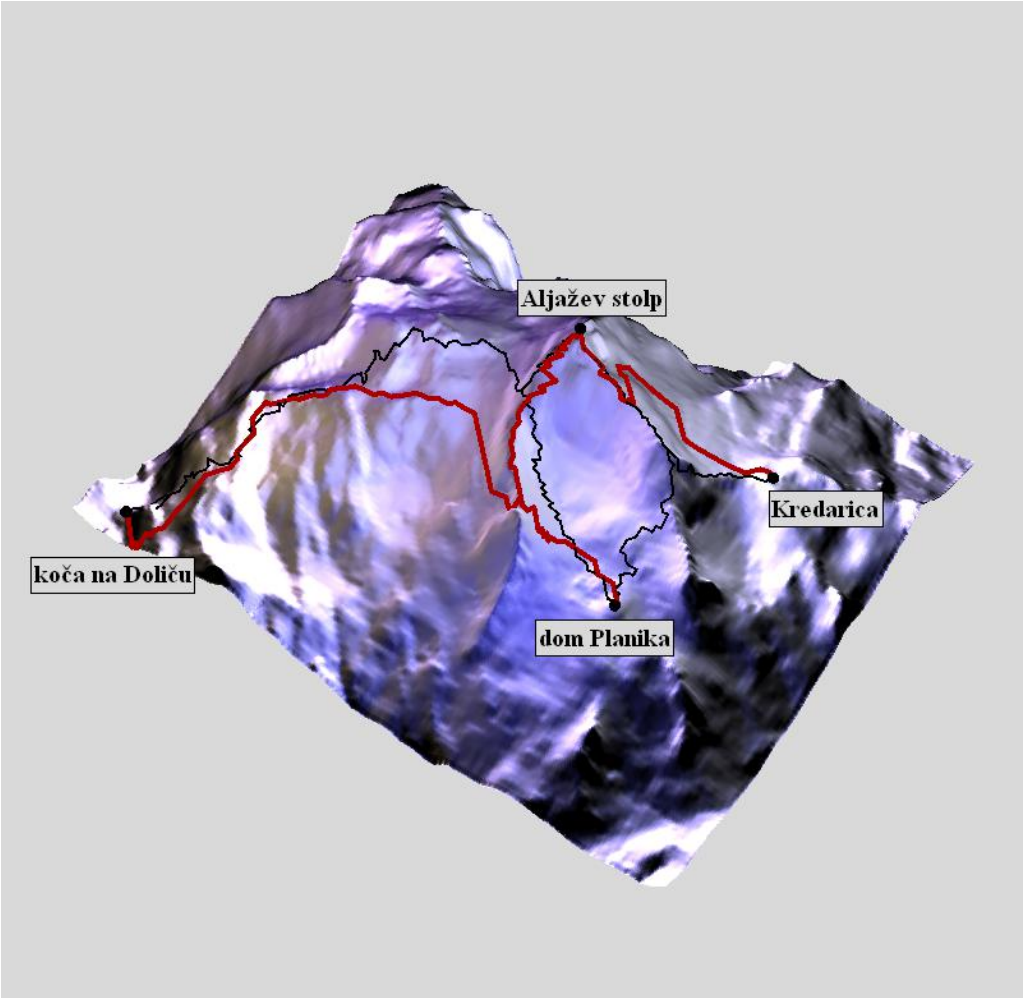
Tabela 3. Poraba energije pri vzponu na Triglav.



Slika 25. Naravne poti (črne črte) in izračunane optimalne poti (rdeče črte) na Triglav.

Ker gre pri Triglavu za velike višine in izredno strm teren, nas je zanimalo, kakšni bi bili rezultati, če bi omejili prečno strmino poti. Dodali smo kriterij, ki utežem

na povezavah oz. delih poti, ki imajo prečni naklon na teren v začetni in končni točki povezave večji od 45 stopinj, določi težo neskončno. Z dodanim kriterijem se optimalna pot izogne zelo strmim predelom in ubere drugačno smer. Na sliki 26 so prikazane optimalne poti, ki smo jih dobili na tak način.



Slika 26. Naravne poti (črne črte) in izračunane optimalne poti (rdeče črte) na Triglav pri dodatnem kriteriju, ki upošteva prečno strmino poti.

Če optimalne poti s slike 26 primerjamo s tistimi s slike 25, opazimo precejšnje razlike. Največjo spremembo je zaznati pri optimalni poti, ki vodi od doma Planika. Z dodatnim kriterijem se precej približa svoji naravni različici preko Triglavske škrbine. Pot od koče na Doliču naredi še večji ovinek po desni strani, nato pa se združi s potjo od doma Planika. Naprej potekata direktno po grebenu.

Podatki o porabi so zbrani v tabeli 4. Po predvidevanjih opazimo upad prihranka energije, ki je posledica dodatnih pogojev in s tem „slabših“ optimalnih poti.

Izhodišče	Pot	Poraba (kJ)	Prihranek (kJ)	Razlika (%)
dom Planika (čez Triglavsko škrbino)	naravna optimalna	4677,78 2238,44	2439,34	52,15
dom Planika (čez Mali Triglav)	naravna optimalna	3154,53 2238,44	916,09	29,04
Kredarica (čez Mali Triglav)	naravna optimalna	4682,56 2391,08	2291,48	48,94
koča na Doliču (čez Triglavsko škrbino)	naravna optimalna	4950,75 3673,80	1276,95	25,79

Tabela 4. Poraba energije pri vzponu na Triglav pri dodatnem kriteriju, ki omeji prečno strmino poti.

Podobno kot v primeru Šmarne gore nas je zanimalo, kako se spremeni poraba energije, če zmanjšamo število interpolacijskih točk in če se spremeni oblika interpolacijske ploskve.

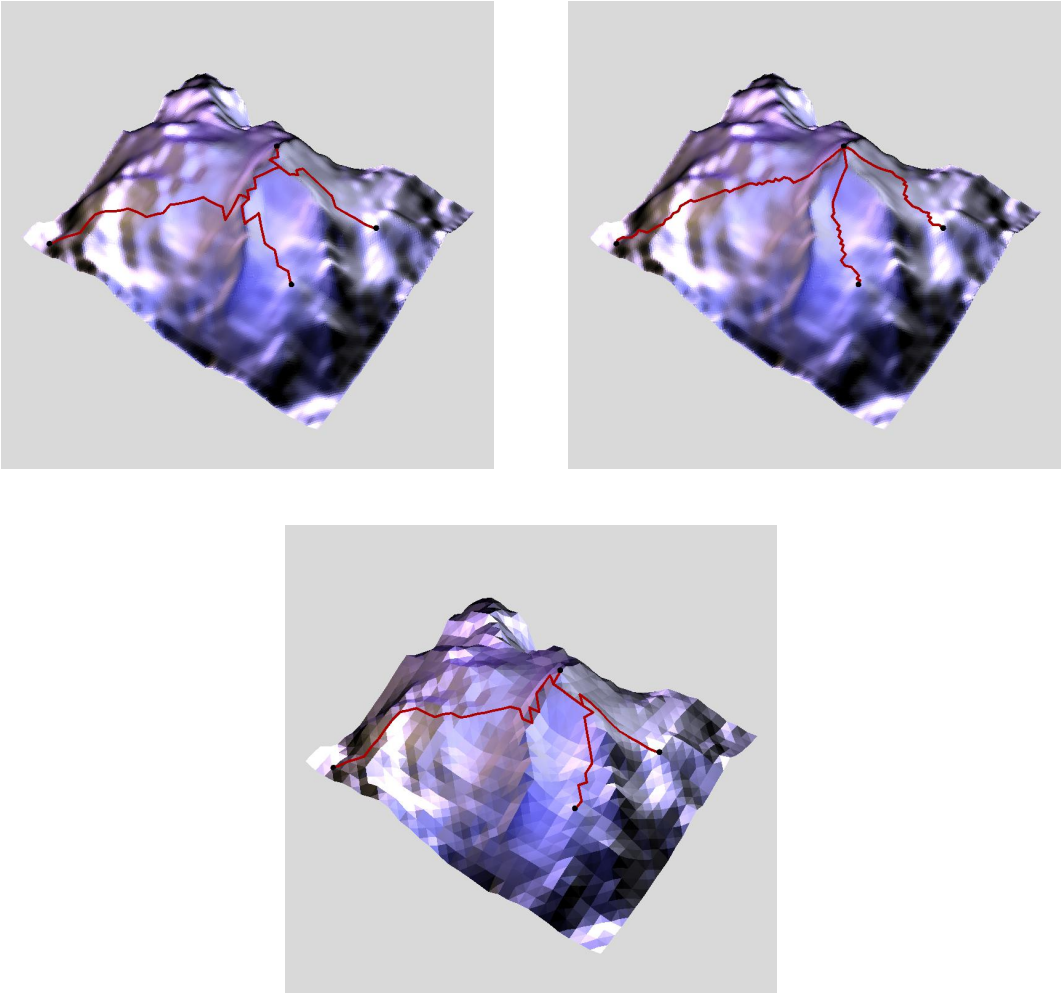
Ponovno primerjajmo vse tri optimalne poti, dobljene na treh različnih ploskvah. Prvi stolpec v tabeli 5 označuje rezultate, ki se nanašajo na grobo mrežo podatkov, v drugem stolpcu so rezultati, izboljšani z enim korakom subdivizije, v tretjem pa tisti, ki so izračunani na podlagi odsekoma linearne ploskve.

V primeru Šmarne gore so rezultati, zbrani v tabeli 2, potrdili naša predvidevanja, da dobimo najbolj točne podatke z uporabo kubičnega zlepka. En korak subdivizije je natančnost še izboljšal, medtem ko je uporaba linearnega zlepka dala precej slabše izračune.

Pričakovali smo, da se bo podobno izkazalo tudi v primeru Triglava. Pri polinomskem zlepku se zmanjševanje števila interpolacijskih točk naj ne bi odražalo na obliki terena in obliki optimalnih poti tako hitro kot pri linearnem zlepku. Podatki zbrani v tabeli 5 so torej zavajajoči, saj smo pri zelo razgibanem reliefu Triglava z linearno aproksimacijo očitno dobili teren s precej manj strmimi deli kot pri polinomski interpolaciji. Posledično je poraba energije pri vzponu po teh poteh manjša. Enako se odrazi pri postopku subdivizije, ki kljub na videz podobnim potem (slika 27) zmanjša porabo energije za kar trikrat.

Izhodišče	Groba mreža	Po enem koraku subdivizije	Odsekoma linearna ploskev
dom Planika	2427,98	790,59	2040,01
Kredarica	2180,12	751,51	1782,53
koča na Doliču	3618,74	1519,37	3135,40

Tabela 5. Poraba energije (v kJ) pri vzponu na Triglav glede na vrsto interpolacijske ploskve.



Slika 27. Optimalne poti na Triglav na grobi mreži podatkov (zgoraj levo), po enem koraku subdivizije (zgoraj desno) in na odsekoma linearni ploskvi (spodaj).

8. Zaključek

Pri pisanju dela diplomskega seminarja, katerega cilj je iskanje optimalne poti na goro, smo precej časa posvetili algoritmu, ki to nalogo opravi. Programsko kodo je bilo potrebno prilagoditi novim podatkom za goro, ki smo jo izbrali. Dodanih je bilo nekaj dodatnih kriterijev, saj so strmine pri Triglavu dosti večje od tistih pri Šmarni gori. Soočili smo se z več težavami, ki pa nam jih je tekom dela uspelo več ali manj odpraviti. Glavna težava, ki je delno ostala neodpravljena, je časovna zahtevnost algoritma. Ker smo želeli, da bi bili dobljeni rezultati čim bolj točni, slike reliefa in poti pa čim bolj natančne, smo vzeli veliko množico vhodnih podatkov. To pa je posledično povzročilo počasno delovanje nekaterih delov algoritma. Čas določanja optimalne poti in računanja porabe energije, potrebne za vzpon, je odvisen od števila interpolacijskih podatkov. Pri najbolj fini mreži točk, ki smo jo uporabili, smo na rezultate čakali več kot en dan.

Prva stvar, ki se izkaže za zelo zamudno, je izračun porabljene energije. Pri tem je namreč potrebno izračunati integral za vsako izmed povezav v uteženem grafu. To bi lahko pospešili z dinamičnim računanjem uteži, saj precej uteži pri optimizaciji niti ne potrebujemo. Drugi problem pa nastane, ko poženemo Dijkstrov algoritem

za iskanje najcenejše poti. Domnevamo, da bi se (vsaj časovno) bolje odrezal kakšen meta-hevrističen problem, kot so npr. kolonije mravelj (več v [2]).

Literatura

[1] I. N. Bronstein, K. A. Semendjajev, G. Musiol in H. Mühlig, *Matematični priročnik*, popravljena izdaja, 1. natis, Tehniška založba Slovenije, 2009.

[2] M. Dorigo in T. Stützle, *Ant Colony Optimization*, MIT Press, London, 2004.

[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 3rd ed., Academic Press, Inc., Boston, 1993.

[4] G. Jaklič, *Krivulje in ploskve v računalniško podprtem geometrijskem oblikovanju*, verzija 28. 2. 2011, [ogled 7. 5. 2013], dostopno na `http://www.fmf.uni-lj.si/~jaklicg/CAGD.pdf`.

[5] G. Jaklič, T. Kanduč, S. Praprotnik in E. Žagar, *Z najmanj truda na Šmarno goro!*, Obzornik mat. fiz. **59** (2012) 1–10.

[6] G. Jaklič, T. Kanduč, S. Praprotnik in E. Žagar, *Energy Minimizing Mountain Ascent*, Journal of Computational and Applied Mathematics **235** (2011) 2758–2765.

[7] T. Kanduč, *Makro elementi*, diplomsko delo, Fakulteta za matematiko in fiziko, Univerza v Ljubljani, 2009.

[8] J. Kozak, *Numerična analiza*, DMFA - založništvo, Ljubljana, 2008.

[9] M. J. Lai in L. Schumaker, *Spline functions on triangulations*, Cambridge University Press (2007).

[10] M. Llobera in T. Sluckin, *Zigzagging: Theoretical insights on climbing strategies*, J. Theo. Bio **249** (2007) 206–217.

[11] B. Plestenjak, *Uvod v numerične metode (matematika)*, verzija 5. 10. 2012, [ogled 17. 5. 2013], dostopno na `http://http://ucilnica.fmf.uni-lj.si/pluginfile.php/5666/mod_resource/content/1/Skripta_UVNM_051012.pdf`.

[12] T. Schröter in M.-N. Glöckner, *How to Climb a Mountain? Simulating efficient ways to the mountain top*, ECMI Newsletter **46** (2009).

[13] Wolfram Research, Inc., *Mathematica*, Version 8.0, Champaign, IL, 2010.